TUTTO QUELLO CHE VORRESTE SAPERE SUL CALCOLATORE

e che non avete mai osato chiedere

Joseph M. Vles - parte prima

ZEROUNO

Arnoldo Mondadori Editore

Tutto quello che vorreste sapere sul calcolatore
e che non avete mai osato chiedere

Tutto quello che vorreste sapere sul calcolatore

e che non avete mai osato chiedere

Traduzione italiana
a cura di
Gerolamo Drei

Arnoldo Mondadori Editore

© 1981 AMACOM A division of American Management Associations, New York
© 1982 Arnoldo Mondandori Editore S.p.a. Milano

Questo volume è stato impresso nel mese di gennaio 1982
presso la Lito 3 - Cologno M. (Milano)
Stampato in Italia - Printed in Italy

Indice

Introduzione 7

Glossario 17

L'hardware 41

Il software 79

Introduzione

Fin dalla loro comparsa sulla scena tecnologica i calcolatori sono stati considerati come creazioni della stregoneria, accessibili soltanto a pochi iniziati. Si tratta di un fatto estremamente negativo, poiché non esiste alcun motivo per associare gli elaboratori alla magia. I calcolatori, infatti, possono essere spiegati e capiti.

Dato il loro ruolo sempre maggiore nella nostra vita e l'influenza crescente sulle nostre attività, le nostre case e la nostra capacità di comunicare è importante rendersi conto degli attuali sviluppi tecnologici. In realtà che cosa sono i calcolatori, che cosa possono o non possono fare?

Il mio tentativo si rivolge ad una vasta gamma di lettori. Se al momento non sapete proprio nulla dei calcolatori, vedrete che dopo aver letto questo libro ne saprete molto di più; anche gli uomini d'azienda – che spesso vengono a contatto con i calcolatori, ma vorrebbero saperne di più per trarne vantaggi immediati – troveranno in questo libro utili riferimenti. Ed ancora un altro gruppo di lettori, ai quali questo libro sarà utile, è rappresentato da coloro i quali pensano di entrare nel campo dei computer, in quanto fortemente motivati, ma non sanno quello che ciò comporta.

I calcolatori che vengono trattati in questo libro sono quelli digitali, *apparecchiature cioè che vengono utilizzate in tutte le applicazioni commerciali. Questo tipo si differenzia dal calcolatore denominato* analogico. *I calcolatori analogici misurano i processi negli impianti di lavorazione, a seguito delle letture che*

ottengono da esplorazioni all'interno di un processo di fabbricazione. L'industria vetraria, ad esempio, può utilizzare calcolatori analogici per regolare il flusso e le proporzioni dei prodotti chimici per tutta la durata di un processo per la produzione del vetro.

La parola analogia *viene definita come "il rapporto di somiglianza fra due oggetti, o di un oggetto con un altro consistente nella somiglianza non degli oggetti stessi, ma di due o più attributi, circostanze o effetti". Un calcolatore analogico crea un'analogia con un problema e misura le risposte. In effetti, anche un regolo calcolatore è un calcolatore analogico. I risultati di un calcolatore analogico hanno sempre una buona approssimazione, di norma, di sole due o tre cifre.*

Oggi i calcolatori digitali vengono impiegati indifferentemente da piccole e grandi società e svolgono funzioni tipiche nelle aree del personale, dell'inventario, delle applicazioni finanziarie, dai sistemi retributivi alla preparazione dei bilanci. I calcolatori, inoltre, vengono spesso impiegati per predire *le esigenze future e i margini di profitto sulla base di fatti noti. Ad esempio, un'acciaieria può impiegare il calcolatore per determinare la mescolanza migliore di materie prime e prodotti chimici in un processo per la produzione dell'acciaio. Il calcolatore considererà le esigenze connesse con la produzione di specifici tipi di acciaio, le combinerà con le materie prime disponibili e determinerà la mescolanza che soddisfi quelle esigenze. Queste operazioni saranno tese ad ottenere il miglior rapporto costi-benefici.*

Una società di costruzioni, o una fabbrica di prodotti destinati a usi molto differenziati, possono impiegare calcolatori per determinare l'attuale disponibilità di risorse, sia in termini di manodopera sia di materiali. Il calcolatore tenterà di associare le risorse disponibili ai termini di scadenza ricordando *l'esigenza di portare a termine certe lavorazioni nell'ambito del processo produttivo, prima di dare inizio ad altre.*

I risultati forniti sono fotografie del futuro *che consentono, alla gestione, di pianificare oculatamente e di determinare le aree di indirizzo dei prodotti. I calcolatori inoltre, vengono spesso impiegati per la determinazione dei costi ipotizzati per la fabbricazione di beni, così da permettere all'impresa di disporre di una buona valutazione dei costi contingenti sostenuti nella produzione di nuovi prodotti. Ciò consente, alla gestione, di definire prezzi redditizi e competitivi nello stesso tempo.*

La modellistica finanziaria è un ennesimo settore di valido utilizzo dei calcolatori, in ambito industriale. Le imprese, infatti, possono definire la propria posizione finanziaria sul mercato in prospettiva per molti anni riuscendo a prevedere certe situazioni e fornendo poi queste informazioni al calcolatore, unitamente al maggior numero di fatti specifici noti che possono essere acquisiti. Ad esempio, un'industria che produce cioccolatini può essere molto interessata nell'individuare l'effetto che avrebbe sui profitti un aumento del prezzo del cacao. Questi programmi di modelli consentono alle industrie di giocare al gioco del "e se". *Ad esempio l'industria cioccolatiera potrebbe decidere di comperare un nuovo macchinario per la fabbricazione dei cioccolatini. Giocando al gioco del* "e se" *con il suo calcolatore, può determinare se l'investimento di capitale per tale nuovo macchinario è giustificato, alla luce di tutte le altre considerazioni sulla produzione, sul marketing e sui costi nel periodo, ad esempio, dei prossimi dieci anni.*

I calcolatori sono masticanumeri. *Vale a dire che sono capaci di eseguire calcoli complessi ed estensivi in un periodo di tempo incredibilmente breve. La modificazione di una sola cifra in un grande calcolo di costi significa che centinaia, e spesso migliaia, di altri numeri dovranno cambiare. Nessun calcolo effettuato dal calcolatore è* magico, *in quanto voi stessi potreste fare altrettanto con carta e matita: la differenza sta nel fatto che, molto probabilmente, quando la vostra* previsione *si sarà concretizzata, sarà passato così tanto tempo da far diventare "storico" il vostro calcolo.*

E questo è il motivo che giustifica l'esistenza dei calcolatori.

Il calcolatore non compie nulla di diverso da ciò che si può fare a mano, ammesso che se ne abbia il tempo. Qualche volta la differenza di tempo fra una soluzione di un dato problema elaborata dal calcolatore ed una manuale può essere misurata in minuti, ore, giorni, ma in situazioni complicate essa potrebbe essere di anni e addirittura di una generazione.

L'enorme accelerazione dello sviluppo e dell'acquisizione tecnica negli ultimi 20 anni è stata possibile solo in virtù della velocità dei calcolatori. Senza calcolatori non avremmo messo piede sulla luna, la medicina non sarebbe al punto in cui oggi si trova, nè saremmo in grado di predisporre il calendario del campionato di calcio.

Sfortunatamente i calcolatori sono anche la causa di tanta spersonalizzazione della società odierna. Gli esperti di calcola-

tori hanno deciso molto presto che l'identificazione della gente per mezzo dei numeri è molto più esatta dall'identificazione per nome. Uno dei nomi più diffusi in Italia è Paolo Rossi. È facile, quindi, confonderli tra loro. Con il calcolatore è possibile assegnare un numero ad ognuno di loro, così da poterli identificare e riconoscere immediatamente. Lo stesso vale per l'assistenza socio-sanitaria, per le patenti di guida, per le carte di credito. I numeri che ci identificano nel settore privato ed in quello pubblico sono senza fine. Anche se sotto certi aspetti, questo metodo si è rivelato per noi efficace e vantaggioso, è indubbio che ha portato ad una specie di spersonalizzazione dell'individuo, negando a volte quell'attenzione e quel rispetto che eravamo abituati a richiedere.

Un serio svantaggio del gioco dei numeri *sta nel fatto che gli esperti di calcolatori si sono talmente abituati ai numeri ed alle schede perforate che spesso è estremamente difficile per l'utente interrompere questo processo e ottenere una risposta* umana *alle domande. In alcuni Paesi stranieri si verifica che, in caso di controversie sui pagamenti tra fornitore e consumatore, derivate da una mancata comprensione delle parti per differenti linguaggi usati, vengono utilizzati interventi* di tipo umano. *Un altro modo per ottenere attenzione personale – ad esempio da parte dell'azienda del gas, di un grande magazzino, o dall'istituto che ha fornito la carta di credito – è quello di "spillare, piegare, mutilare" la carta di credito stessa.*

Generalmente la cattiva annotazione scritta sulla scheda non sarà mai letta da nessuno (anche se siete riusciti a scriverla in maniera leggibile cercando di evitare le perforazioni). La vostra scheda viene raccolta e ordinata assieme a migliaia di schede simili, immessa in una macchina lettrice di schede collegata ad un calcolatore che a sua volta la interpreterà. *Per il calcolatore ciascun foro della scheda ha un significato. Alcune combinazioni di fori formano delle lettere, altre formano numeri e le perforazioni tutt'insieme identificano voi, il numero del vostro conto corrente e il vostro saldo debitore. Spesso, dopo aver spedito la scheda perforata con allegato l'assegno, la scheda stessa subirà ulteriori perforazioni. Queste indicano la somma che avete pagato e, nel contempo, nuove perforazioni annullano qualsiasi annotazione da voi scritta sulla scheda.*

Quando la scheda passa alla macchina ogni foro si traduce in impulsi elettronici e questi impulsi aggiornano nel calcolatore i dati che si riferiscono a voi. Questo processo di immagazzi-

namento e aggiornamento delle informazioni può essere paragonato all'incisione di una canzone su un nastro. La canzone viene registrata una volta, proprio come le informazioni sul vostro conto corrente vengono create una volta nel calcolatore. Si può suonare la canzone molte volte, così come più volte potete chiedere al calcolatore le informazioni che vi riguardano. Queste informazioni possono essere aggiornate *registrando una nuova canzone su una vecchia. Il calcolatore aggiorna la vostra situazione in maniera analoga quando cambia qualcosa che vi riguarda, come nel caso dell'effettuazione di un pagamento.*

Se mutilate *la vostra scheda, interrompete il processo automatico, perché la scheda non può essere più letta dall'apparecchiatura. Il pagamento, allora, dovrà essere registrato manualmente ed è possibile che riceviate risposta agli scarabocchi fatti sulla scheda. Così avrete ancora una identità precisa, ma solo fino a che il sistema automatico per il trasferimento di fondi non entrerà in funzione in tutto il territorio nazionale. Quel giorno non vedrete neppure più le vostre fatture, i pagamenti saranno trasferiti automaticamente, via calcolatore, dalla vostra banca alla banca emittente la fattura senza ricorso a lavoro d'ufficio o a schede perforate.*

I calcolatori sono realizzati in forme e dimensioni diverse. Come per le automobili, la competitività fra produttori è elevata; indubbiamente avrete sentito parlare di tutti i tipi delle varie apparecchiature. La gente parla di Ibm 370/168, Dec-10, 11/70 e Sistema 32 in modo tale da farvi sembrare che vi sia così tanto da sapere ancor prima che vi impadroniate di un minimo di nozioni. Da ciò spesso nasce un atteggiamento di sfiducia nella possibilità di acquisire qualche conoscenza in questo campo. I fanatici di vetture sportive usano lo stesso linguaggio quando parlano delle loro macchine; ma per la maggior parte di noi una macchina è una macchina il cui scopo effettivo è di portarci da qui a lì.

Per noi è facile pensare alle macchine in questo modo, ma per qualche ragione il nostro punto di vista sui calcolatori è diverso. Guardiamo ai calcolatori come se fossero magici e crediamo di non avere altra scelta se non quella di aver cieca fiducia nella scheda perforata del calcolatore.

È originata da un calcolatore e pertanto non può essere sbagliata, non è così?

Errore. Gli esperti di computer godono di questa aureola di onnipotenza e scivoleranno nel linguaggio dell'elaborazione

dati *ogni volta si presenti l'occasione per confondervi un po' di più. In effetti, gran parte di questo discorso dell'elaborazione dati nasconde una scienza lineare. I calcolatori sono soltanto delle macchine mute.*

Mi preme porre l'accento su questo aspetto: i calcolatori non *pensano;* non *sono in grado di prendere decisioni in proprio. I calcolatori fanno solo ciò che i programmi dicono loro di fare e se i programmatori dicono a un calcolatore di fare cose sbagliate, il calcolatore le farà fedelmente, come altrettanto fedelmente esegue quelle corrette.*

Una persona che ha trattato calcolatori fin dalla loro prima comparsa mi ha raccontato una storia interessante. Alcuni anni fa, quando si dedicava all'analisi dei sistemi per gli utenti dei servizi offerti dalla sua ditta, un'industria meccanica gli chiese di scrivere un programma per la risoluzione di un dato numero di equazioni di meccanica applicata alle macchine. Si incontrò con l'incaricato del sistema e gli pose tutte le domande, per conoscere le risposte, necessarie per poter scrivere il programma, ma ad un certo punto si verificò una situazione di stallo. Se una certa condizione era vera, il programma doveva seguire una linea logica definita; se veniva soddisfatta un'altra condizione, si sarebbe dovuto seguire un'altra linea logica. Emerse però che avrebbe potuto esserci ancora una terza condizione. "Cosa devo fare in questo caso?", chiese quella persona. L'ingegnere ci pensò sopra per un istante e disse: "Bene, è un problema di difficile soluzione. Sa cosa le dico: se il programma giunge a quella condizione, perché non lascia che il calcolatore decida cosa fare? A dire il vero io non so quale dovrebbe essere la risposta".

Questo racconto mette in luce i concetti erronei di molti nei confronti dei calcolatori. Molti di noi sono portati a credere che i calcolatori siano in grado di pensare, ma i calcolatori fanno soltanto ciò che vien detto loro di fare. Quando un calcolatore è programmato per eseguire qualcosa, vi riesce con estrema velocità. Questo vale anche se gli viene dato l'ordine di compiere una qualche azione in modo inesatto.

Il confronto che spesso viene fatto fra calcolatore e cervello umano è ridicolo. È come paragonare una bicicletta ad un Concorde, *in cui il calcolatore è la bicicletta.*

Se mai volessimo imitare tutti i processi del pensiero che avvengono nel cervello umano non cominceremmo certamente dalle possibilità del calcolatore commerciale di oggi, come non

cominceremmo dagli accessori e dalla struttura di una bicicletta per realizzare un aereo supersonico.

Il calcolatore ha due componenti: l'hardware e il software.

L'hardware è il complesso dei componenti fisici del calcolatore; il software è rappresentato da tutti i programmi che ci aspettiamo vengano elaborati dal calcolatore. Talvolta il termine software viene interpretato in senso più lato.

Spesso gli specialisti si riferiscono al software quando parlano di documentazione. Ma, nel nostro caso, il termine software viene riferito rigorosamente ai programmi che riguardano un calcolatore. Disporre solo dell'hardware, senza avere i programmi (software) è lo stesso che avere una vettura senza qualcuno che la guidi. La macchina può apparire bella e potente, è sicuramente costosa, ma non vi porterà da qui a lì. Per rendere operativa la vettura, è necessario un autista.

Come ci sono buoni e cattivi autisti, così, nei calcolatori, ci sono software buoni e cattivi. I cattivi autisti provocano incidenti; i cattivi software danno risultati inesatti o corretti solo in parte. L'hardware di un calcolatore può anche essere paragonato ad un contabile addetto al libro paga, ed il software alle istruzioni di cui quell'impiegato ha bisogno per perfezionare il processo relativo alle retribuzioni. Il contabile è pronto, disponibile e in grado di assolvere il compito dei calcoli propri del libro paga, ma è necessario che disponga delle istruzioni che indicano come il compito deve essere assolto.

I produttori di hardware continuano a migliorare la velocità con la quale le loro apparecchiature effettuano i calcoli.

Continuano anche ad aumentare le prestazioni dei loro calcolatori, in modo da risolvere problemi sempre più complessi; ma sta agli specialisti di software, agli analisti, ai programmatori utilizzare queste maggiori possibilità per ottenere una produzione con risultati il più corretti possibile. In ultima analisi i calcolatori sono validi nella misura in cui è valido il loro software.

L'informatica può essere suddivisa in cinque distinte categorie. La prima è quella con la quale tutti noi veniamo a contatto.

È denominata elaborazione dati e include programmi di fatturazione, sistemi di crediti esigibili, di crediti dovuti e così via. Molti programmi di elaborazione dati trattano la creazione, l'aggiornamento e il riporto delle informazioni. Anche lo spoglio dei dati fa parte dell'elaborazione.

Grandi segmenti delle informazioni sono ridotti *in rapporti, allo scopo di fornire solo una sinossi di parte delle informazioni o di rilevare condizioni eccezionali nell'ambito delle informazioni stesse. La categoria dell'elaborazione dati, di norma, implica anche l'esecuzione di alcuni calcoli, ma la parte di calcolo dell'elaborazione dati è meno estesa della parte di trattamento delle informazioni.*

La seconda categoria dell'informatica tratta estesi calcoli numerici. Nonostante possano entrare in gioco anche altre informazioni, lo scopo fondamentale dei programmi di calcolo numerico è quello di ottenere risposte ai problemi matematici. I programmi di calcolo numerico sono scritti in genere per risolvere formule di ingegneria o matematica: ad esempio per determinare le traiettorie spaziali dei satelliti.

La terza categoria dell'informatica è denominata elaborazione logica. Si tratta di un segmento ancora in via di sviluppo e che costituisce l'area nella quale i calcolatori sembrano essere intelligenti. *I sistemi che appartengono a questa categoria tentano di imitare le azioni del cervello, ma solo attraverso un insieme di passi logici e predeterminati. Ne sono esempi i programmi del gioco degli scacchi, quelli che individuano campioni (video e audio) e quelli infine che traducono frasi da una lingua a un'altra.*

La quarta categoria riguarda i controlli di processo. In quest'area le informazioni vengono inserite nel calcolatore e utilizzate per determinare le azioni successive da intraprendere. Un esempio di questa categoria è quello della determinazione delle proporzioni di prodotti carburanti e correttivi da aggiungere al metallo liquido in un procedimento di fabbricazione dell'acciaio, sulla base di un rapporto sullo stato attuale *fornito da un calcolatore, che confronta detto stato del procedimento con il risultato finale voluto.*

La quinta area nella quale i calcolatori esercitano una funzione sempre più dominante è quella dell'elaborazione di parole e testi negli uffici. I calcolatori vengono utilizzati come sussidio per scrivere lettere, per immagazzinare informazioni, che di solito venivano immagazzinate in armadi archivio, e per produrre materiale pronto per la creazione di composizioni dattilo o su microfiche. *L'utilizzazione dei calcolatori per queste operazioni è spesso denominata* ufficio del futuro, *nonostante dovremmo già cominciare a definirlo* l'ufficio del momento attuale. *Il costo dell'acquisizione di un piccolo calcolatore per*

ufficio con software per l'elaborazione di testi oggi si ammortizza velocemente, per cui l'apparecchiatura comincia a pagare dividendi considerevoli, in termini di risparmio di costi e tempi, subito dopo l'acquisto. Anche il manoscritto originale di questo libro è stato creato su un calcolatore con l'ausilio di software per l'elaborazione dei testi.

Uno dei principali vantaggi che derivano dal trattamento di testi (word processing) è che la modifica di una parte del testo, o lo spostamento del manoscritto in altra posizione vengono attuati rapidamente e facilmente dal calcolatore.

Tutti gli adattamenti necessari e le variazioni avvengono con il software per l'elaborazione dei testi; anche la rinumerazione delle pagine, se necessaria, viene effettuata contemporaneamente. I sistemi di word processing possono trattare la marginatura a sinistra, l'impaginatura a bandiera sul lato destro, varie lunghezze della carta, centratura e caratteri maiuscoli *del testo, note a pie' di pagina, tabelle, per citare solo alcune funzioni. Un sistema di elaborazione, accoppiato con uno per la redazione di testi, che tratti la creazione e le modifiche di materiale testuale nell'apparecchiatura di memorizzazione del calcolatore, è estremamente utile nel lavoro d'ufficio.*

Gli elaboratori di testi, normalmente, stampano le informazioni su terminali con tastiera ad alti livelli di qualità, e le lettere e i documenti prodotti sembrano dattiloscritti a mano.

*La grafica è un altro segmento importante dell'impiego di calcolatori nell'*ufficio del futuro. *Un terminale grafico in ufficio può stampare dati sotto forma illustrata quali curve e diagrammi a barre. Molte imprese utilizzano la grafica dei dati per illustrare le informazioni oggetto di relazioni.*

Oggi sono disponibili anche diagrammi a colori.

Glossario

Questo glossario contiene tutti i termini tecnici usati nel testo. Tutte le volte che è stato possibile, viene fornita la traduzione italiana più vicina al significato del vocabolo originale in inglese.

accesso casuale / random access Accesso ai dati senza tener conto dei dati non pertinenti conservati nell'unità o nel *file* che precedono i dati richiesti.

accesso sequenziale / sequential access Lettura e/o scrittura di dati da e su disco o nastro magnetico in modo sequenziale.

aggiornamento / update Sovrapposizione di nuove informazioni a dati esistenti.

aggiornamento dell'archivio / file updating Operazione di modifica delle informazioni in un archivio dati.

alfanumerico / alphanumeric Tutte le lettere dell'alfabeto e tutti i caratteri speciali. Termine riferito anche ai numeri, se considerati come caratteri.

Algol / Algol Linguaggio di programmazione diffuso in Europa. Acronimo di **A**lgorithmic **L**anguage: linguaggio algoritmico.

algoritmo / algorithm Procedimento o schema per la risoluzione di un problema con un numero finito di passaggi che di frequente comporta la ripetizione di un'operazione.

ampliamento / hang-on Apparecchiatura periferica o addizionale a complemento della configurazione dell'hardware originale di un calcolatore.

analisi / analysis Identificazione dei fatti, e delle loro interrelazioni, attinenti alla completa comprensione di un problema, allo scopo di trovare una soluzione soddisfacente.

analista / analyst Specializzato che trasferisce un problema da risolvere su un'unità che espone tutte le interrelazioni logiche necessarie, tali da consentire la risoluzione del problema da programmare.

annidato / nested Termine riferito all'ipotesi che in un programma FORTRAN si abbia un *loop* all'interno di un *loop*.

ANSI American National Standards Institute/Istituto Nazionale Americano per gli Standards.

apertura di sessione / sign-on Procedura di inserimento che stabilisce il contatto fra utente e calcolatore.

APL Linguaggio di programmazione. Acronimo di **A P**rogramming **L**anguage.

applicazione / application Soluzione di un problema mediante approccio computerizzato.

approccio di gruppo / team approach Programmazione di un sistema da parte di più persone.

archivio / file Una o più registrazioni di informazioni immagazzinate come unità. Un archivio può essere una base dati o un sottomultiplo di una base dati.

archivio dati / data file Termine spesso usato indifferentemente per base dati. Le basi dati, peraltro, sono spesso costituite da più archivi dati.

ascendente / bottom-up Approccio all'impostazione di un programma dove ogni funzione viene scritta quando necessario, senza aver prima programmato funzioni fittizie di prova del flusso del programma e della sua logica.

assegnazione / assignment Inserimento di un valore in un indirizzo della memoria.

background / background Programma di un computer effettivamente in corso senza bisogno di alcun intervento, mentre è in atto anche un programma successivo, o interattivo.

banda / paper tape Striscia di carta perforata. Un'unità, collegata con un terminale, *legge* le perforazioni trasformandole, a sua volta, in impulsi elettrici e viceversa, utilizza impulsi elettrici per scrivere informazioni (vale a dire perforazioni) sulla banda.

base dati / data base Insieme organizzato di dati accessibile mediante un programma per effettuare aggiornamenti, cancellazioni, rapporti.

base 2 / base 2 Sistema binario, nel quale ogni unità elementare può assumere uno o due valori "on" o "off"

Basic / Basic Linguaggio di programmazione. BASIC è l'acronimo di **B**eginners **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode. (Codice per principianti d'istruzione simbolica generale).

batch differito / deferred batch Esecuzione di programmi in tempi differiti.

batch processing / batch processing Elaborazione differita di programmi.

binario / binary vedere *base 2.*

bit / bit La più piccola unità di una parola in grado di memorizzare un valore.

blocco / block Unità specifica di memorizzazione su insieme di dischi magnetici. Termine usato anche per indicare la sezione autonoma di un programma.

bobina / reel Bobina per nastro.

byte / byte Termine non scientifico attribuito ad un carattere di memorizzazione. Il byte si compone di un certo numero di bit: otto, secondo la generica accezione.

cablato / hardwired Unità collegata direttamente a un calcolatore senza il tramite di un collegamento telefonico.

calcolatore / computer Macchina che compie operazioni ad alta velocità, impiegando istruzioni pre-scritte. Le istruzioni pre-scritte (programmi) possono essere modificate facilmente, in modo da risolvere problemi diversificati.

calcolatore analogico / analog computer Calcolatore che effettua i calcoli costruendo un'analogia e misurando le risposte sotto forma di voltaggi, distanze e così via, anziché calcolare le risposte stesse in modo matematicamente preciso. Vedere *calcolatore digitale.*

calcolatore digitale / digital computer Calcolatore che per il calcolo dei risultati utilizza sequenze on-off di impulsi elettronici. I calcolatori digitali trovano impiego nelle applicazioni commerciali, dall'elaborazione di testi alla modellistica finanziaria.

campo / field Suddivisione di una registrazione in un archivio dati.

capo gruppo / team leader Manager di un gruppo di programmatori che lavora allo stesso sistema.

carattere / character Lettera, numero o simbolo speciale.

card deck / card deck Un certo numero di schede perforate sovrapposte.

carico / load Entità delle risorse che un calcolatore impiega per eseguire una particolare funzione.

cassetta / cassette Unità autonoma a nastro magnetico impiegata per trasmettere dati a e da computer. Normalmente annessa ad un terminale a distanza.

cassetta per lettere / mailbox Sinonimo di *indirizzo, ubicazione della memoria, parola* e così via.

centro servizi / service bureau Ditta specializzata, che soddisfa le esigenze di elaborazione della clientela, ricorrendo all'elaborazione differita dei programmi.

chiusura di sessione / sign-off Procedura di disinserimento che interrompe il contatto fra utente e calcolatore.

COBOL Linguaggio di programmazione. Acronimo di **C**ommon **B**usiness **O**riented **L**anguage (Linguaggio di programmazione per applicazioni di carattere commerciale).

codice / code Sezioni di un programma, ovvero l'intero programma di un calcolatore.

codice del programma / program code Sequenza logica di istruzioni all'interno di un programma.

codificazione / encoding Trasposizione (di un'informazione o di un dato) da caratteri a rappresentazione codificata.

collazionatrice / collator Macchina per mezzo della quale schede di varie provenienze vengono collazionate in un determinato ordine.

collegamento / connection Collegamento fra utente e cal-

colatore effettuato, di norma, a mezzo terminale collegato a un telefono e a un accoppiatore.

combinazione / merge Combinazione di dati provenienti da due o più fonti.

compilatore / compiler Programma di sistema che trasforma le istruzioni, scritte in linguaggio ad alto livello, nel linguaggio del calcolatore.

configurazione / configuration Termine usato per indicare le apparecchiature, primarie e periferiche, di cui si compone un sistema di elaborazione dati.

configurazione delle registrazioni / record layout Segmentazione di una registrazione nelle sue componenti denominate campi.

connessione / dial-out Inizio del collegamento fra un calcolatore e un terminale di utenza, originato dal calcolatore stesso. **Dial-up** è il termine antitetico del precedente. Infatti nel dial-up è l'utente che inizia il collegamento.

consecutivo / consecutive L'uno dopo l'altro. Termine riferito alle istruzioni di un programma eseguite consecutivamente, a meno che non sia indicato diversamente nel programma stesso.

conservazione fuori linea / off-line storage Vedere *nastro magnetico*

contenuto / content Valore situato in una memoria.

controllo di validità / validity check Controllo avente lo scopo di determinare se il contenuto di una variabile è entro un certo campo, o se ha un valore specifico.

conversione / conversion Lettura del codice di un programma di un calcolatore in un altro linguaggio, ad esempio da COBOL a FORTRAN.

correzione / debugging Processo per la correzione della sintassi e della logica di un programma.

costante / constant Valore invariabile.

CPU Unità centrale di elaborazione. Acronimo di: **C**entral **P**rocessing **U**nit.

dati / dato Informazioni conservate in un magazzino del calcolatore.

dati contigui / contiguous data Dati immagazzinati consecutivamente in una memoria.

decodificazione / decoding Trasposizione da codice a caratteri che possono essere interpretati.

design / design Progressione logica degli eventi di un programma per dare soluzione a un problema, dopo che questi eventi sono stati codificati in linguaggio di programmazione.

designer / designer Specialista che determina il linguaggio e la disposizione da impiegare in un programma.

designer di sistemi / system designer Vedere *designer*

diagramma di flusso / flowchart Rappresentazione grafica di un sistema o di un programma.

diagramma di flusso dettagliato / detailed flowchart Rappresentazione illustrata di tutti gli eventi di un programma.

diagramma di flusso di sistemi / system flowchart Vedere *flowchart*.

difetto / bug Errore di programmazione.

digitale / digital Espressione di caratteri mediante cifre del sistema decimale o di altro sistema.

dimensione / dimension Termine riferito, in un programma, a variabili con la stessa denominazione, ma che hanno

indirizzi differenziati. Ad esempio, una variabile che può assumere 3 valori diversi, viene denominata IVAL, ma ciascun valore si distingue dagli altri della serie per la base. IVAL (1), IVAL (2), IVAL (3) nella memoria sono parole distinte e ciascuna contiene un valore suo proprio.

dimensione / size Termine generalmente usato per indicare:
– quantità dell'immagazzinamento che un sistema ha disponibile in linea;
– numero delle parole contenute in una memoria;
– capacità di elaborazione della CPU.

discendente / top-down Approccio alla costruzione di un programma nel quale ciascuna funzione è programmata prima sotto forma scheletrica in modo da verificare con facilità il diagramma di flusso del programma o la logica.

disco / disk Elemento per la memorizzazione di dati, somigliante fisicamente a una pila di dischi.

distruzione del disco / disk crash Distruzione dei dati di un disco dovuta a difetti di fabbrica.

do loop / do loop Circuito che consente ad una sezione del programma di essere eseguita una o più volte. Il numero delle iterazioni dipende da un parametro numerico.

durata del ciclo della memoria / memory cycle time Intervallo di tempo durante il quale un'unità d'informazione può essere ritirata dalla memoria o immessa.

elaborazione dati / data processing Termine usato in senso lato per indicare tutte le sfaccettature delle operazioni di elaborazione. Più esattamente l'elaborazione dati coinvolge l'impiego di calcolatori per la trattazione di grandi quantità di dati con importanza relativamente scarsa del calcolo.

elaborazione della parola / word processing Termine generico riferito alla trattazione del lavoro d'ufficio utilizzando un calcolatore.

Elaborazione distribuita / distributed processing Decentralizzazione dell'utilizzazione di un calcolatore, attraverso una rete di calcolatori con capacità di interfaccia fra unità qualsiasi.

entrata / input Informazione che viene introdotta nella memoria o nel magazzino permanente di un calcolatore.

esadecimale / hexadecimal Sistema numerico in base 16.

etichetta / label Indicazioni poste all'inizio delle istruzioni per un programma. Quando il programma si mette in funzione, il controllo può essere passato alle istruzioni etichettate anche se l'esecuzione non viene effettuata sequenzialmente.

fiche / fiche Foglio di microfilm contenente un notevole numero di pagine in formato molto ridotto.

flusso logico / logic flow Progressione logica, evento dopo evento, delle istruzioni di un programma.

format / format "Griglia" impiegata per l'entrata e l'uscita dei dati.

FORTRAN Linguaggio di programmazione. Acronimo di Formula Translation.

frase / sentence Istruzione a un calcolatore scritta in linguaggio di programmazione. In genere il termine si riferisce al linguaggio COBOL.

funzione / function Sottinsieme logico di un programma.

funzione fittizia / dummy function Sezione del programma di un calcolatore completa, ma solo sotto forma scheletrica.

fuori linea / off-line Apparecchiatura e/o informazioni in atto non facenti parte del sistema operativo di un calcolatore.

gestione servizi / facility management Utilizzazione di calcolatori, programmi, e/o specialisti, offerti in locazione a terzi, da fornitori di servizi su base *time sharing*.

gioco delle parti / role playing Procedura di intervista nel corso della quale l'intervistato si comporta come un analista e l'intervistatore come una persona che ha un problema che deve essere risolto da un computer.

GIGO Acronimo di **G**arbage **I**n, **G**arbage **O**ut. Letteralmente: immondizie dentro, immondizie fuori. Espressione usata per indicare che se si mettono nel calcolatore dati inesatti, si ottengono risposte inesatte.

grandezza della memoria / memory size Numero di parole o indirizzi contenuti in una memoria. A volte è misurata in *byte* (ciascuno pari a otto bit) anziché in *parole*.

griglia background / background mask "Griglia" rappresentata su un terminale video, sotto controllo del programma, per facilitare l'entrata dei dati.

hard-copy / hard-copy Carta o cartoncino per la stampa in entrata e uscita di dati e/o informazioni.

hardware / hardware L'insieme dei componenti fisici di un calcolatore.

housekeeping / housekeeping Cancellazione di dati estranei da un disco. Termine riferito anche alle istruzioni di un programma che assegna i valori iniziali alle variabili.

immagazzinamento-memorizzazione / storage Usato anche il termine immagazzinamento. In genere il termine si riferisce ad insiemi di dischi e/o di nastri magnetici ma può anche riferirsi all'immagazzinamento di nastri di carta e cassette.

immagazzinamento temporaneo / temporary storage Dati immagazzinati nella memoria di un calcolatore.

in linea / on-line Apparecchiatura e/o informazioni in atto facenti parte del – o collegate al – sistema operativo di un calcolatore.

in loco / in-house Termine usato per indicare la disponibilità di un calcolatore in loco.

incidente / crash Inoperatività di un calcolatore dovuta a una fra tante cause. Un incidente non distrugge necessariamente i dati (vedere *disk crash*).

indirizzo / address Posizione della memoria nella quale vengono conservati informazioni e/o dati.

indirizzo della memoria / core location Indirizzo della memoria nel quale è possibile immagazzinare i valori. Sinonimo, fra gli altri termini, di *indirizzo*, *parola* e *cassetta per lettere*.

indirizzo della memoria / memory address Sinonimo di *ubicazione della memoria*, *parola*, *cassetta per lettere*, e così via.

insieme / array Insieme di indirizzi memorizzati (variabili).

insieme / pack Termine comunemente usato come insieme di dischi.

insieme di dischi / disk pack Vedere *disk*.

installazione / installation Il complesso di hardware di cui sono dotati un calcolatore e relative stazioni periferiche. Indica anche l'operazione di messa a punto di un calcolatore nuovo.

interattivo / interactive Termine usato per indicare l'interazione fra calcolatore e utente di un terminale.

interruzione / interrupt Termine riferito di norma a una CPU che svolge più di una operazione in un tempo che sembra essere *quello del momento*. Quando si verifica un'in-

terruzione, l'operazione viene *accantonata* in modo da portare l'attenzione su un'altra operazione.

istruzione / statement Istruzione a un calcolatore, scritta in linguaggio di programmazione.

K Indicativo della quantità 1024, usato come misura della capacità di memoria.

letterale / literal Testo interpretato dal calcolatore alla lettera, cioè non come un'istruzione.

lettrice / card reader Macchina che interpreta le configurazioni dei fori delle schede trasformandole in impulsi elettronici, interpretabili da un calcolatore.

lettura / retrieve Lettura di dati da un magazzino permanente a una memoria.

lettura / read Termine generico riferito a un programma che accede a dati immagazzinati e li trasferisce a una memoria.

lettura di marcature / mark sensing Processo elettrico impiegato molto spesso per *correggere* tests del tipo a risposte multiple.

linea principale / main line Parte di un programma che controlla la sequenza di esecuzione dei vari moduli.

linguaggio / language Insieme di istruzioni relative ad un programma, inteso a dare istruzioni operative al calcolatore.

linguaggio assemblatore simbolico / symbolic assembly language Linguaggio che interagisce direttamente con l'hardware; denominato anche linguaggio a basso livello, in contrapposizione a linguaggi ad alto livello quali il COBOL e il FORTRAN. In questo caso *basso* significa più vicino all'interazione attuale dell'hardware e *alto* significa molto lontano da quel livello.

linguaggio di programmazione / programming language Repertorio delle istruzioni che un calcolatore è in grado di interpretare e elaborare.

linguaggio macchina / assembly language Linguaggio che interagisce direttamente con la macchina; denominato anche linguaggio a basso livello, in contrapposizione a linguaggi ad alto livello. In questo caso, *basso* significa più vicino all'interazione attuale dell'hardware e *alto* significa più distante da quel livello.

livello / level Rappresentazione gerarchica di elementi di dati in sottoinsieme di una registrazione o di una variabile. Termine usato nel linguaggio COBOL.

loop / loop Sezione del programma di un calcolatore che può essere eseguita una opiù volte, in funzione del valore di un parametro numerico.

maschera / mask Forma generata da un calcolatore che può fare da sfondo a informazioni sugli schermi televisivi dei terminali.

maschera / picture *Griglia* impiegata per l'entrata e l'uscita di valori dei dati.

masticanumeri / number cruncher Si dice di un calcolatore in grado di effettuare calcoli a velocità molto elevata.

memoria / memory Unità di cui è dotato un calcolatore, nella quale risiedono durante l'uso programmi e dati.

memoria a nuclei / core Tipo di memoria largamente impiegata nel passato; sostituita dalle memorie a semiconduttori. Attualmente il termine è ancora sinonimo di memoria del calcolatore.

memoria permanente / permanent storage Magazzinaggio di dati su disco o nastro magnetico.

metodo d'accesso / access method Modo con cui si stabilisce l'interazione uomo-calcolatore.

microcomputer / microcomputer Calcolatore molto piccolo, più piccolo di un minicomputer.

microfiche / microfiche Foglio di microfilm contenente un gran numero di pagine di formato molto ridotto.

microsecondo / microsecond Milionesimo di minuto secondo.

minicalcolatore / minicomputer Piccolo calcolatore dotato di tutti i componenti hardware di un grande calcolatore, ma di minori dimensioni. In genere il minicomputer è anche caratterizzato da velocità di elaborazione minore di quella di un grande calcolatore.

millisecondo / millisecond Millesimo di minuto secondo.

modellistica / modeling Applicazione di vari dati d'entrata su una pista a soluzione premisurata, per misurare la varianza d'uscita quando si verifica una modifica di uno o più dati in entrata.

modulo / module Segmento di un programma logico.

nastro di lavoro / scratch tape Nastro magnetico non usato, al momento, per immagazzinare dati che dovrebbero essere memorizzati.

nastro magnetico / magnetic tape Unità di memoria per la conservazione di dati e programmi. Definita anche memoria esterna (off-line storage).

nastro magnetico / tape Mezzo off-line di immagazzinamento di dati. Le informazioni vengono depositate magneticamente sul nastro.

NBS National Bureau of Standards.

nome variabile / variable name Termine per la locazione di una memoria. Si usano anche i termini *parola*, *ubicazione della memoria a nuclei*, *indirizzo*, e così via.

numerico / numeric Dati utilizzabili nei calcoli.

numero frazionario / floating-point number Numero frazionario. Si dice anche numero reale.

numero intero / integer Numero intero.

numero reale / real number Numero frazionario. Vedere anche *floating-point number*.

OCR Vedere *Optical character recognition*

ottale / octal Sistema numerico in base 8.

ordinamento / sort Riordino dei dati in modo da realizzare una sequenza ascendente o discendente.

ordinatrice / card sorter Macchina per mezzo della quale si selezionano le schede in un ordine predeterminato, in base a una o più colonne di informazioni contenute nelle schede stesse.

pacco di schede / deck Termine per indicare un certo numero di schede perforate. Di norma si usa riferito a una certa quantità. Il *deck* può riferirsi al codice di un programma, o ai dati di un archivio.

paragrafo / paragraph Una o più istruzioni del programma COBOL strutturalmente collegate. L'esecuzione del paragrafo comprende tutte le istruzioni di quel paragrafo, a meno che il programma non sia obbligato ad abbandonare il paragrafo dopo aver seguito la procedura del controllo di flusso.

parola / word Sinonimo di *ubicazione della parola, ubicazione della memoria o nuclei, indirizzo, cassetta per lettere*.

parola della memoria / memory word Sinonimo di *indirizzo, ubicazione* e così via.

parole riservate / reserved words Parole che non possono essere usate da un programmatore come nomi assegnati, perché il linguaggio del programma usa queste parole come suoi operatori.

perforatrice / card punchunit Macchina per mezzo della quale si perforano le schede meccanografiche. I fori formano configurazioni che rappresentano dati alfabetici e numerici.

perforazione / keypunching Produzione di schede perforate con memorizzazione dati esterni, mediante l'impiego di una macchina perforatrice.

perno / spindle Perno sul quale si monta un insieme di dischi.

pista logica / logic path Pista di esecuzione che un programma localizza una volta inserito nel calcolatore.

PL/1 / PL/1 Linguaggio di programmazione. PL/1 è l'acronimo di **P**rogramming **L**anguage n. **1** (linguaggio di programmazione n. 1).

plotter / plotter Unità di uscita di un calcolatore che presenta i dati sotto forma di grafici.

potenza di un calcolatore / computer power Le risorse di un calcolatore misurate in termini di dimensioni e velocità operativa.

potenziamento / upgrade Aggiunta di hardware alla configurazione attuale di un calcolatore.

procedura / procedure Parte segmentata di un programma.

fail-safe procedure / fail-safe procedure Copiatura su nastro magnetico di dati e programmi memorizzati su disco.

processore / processor Abbreviazione di *central processing unit*

programma / program Complesso di istruzioni impartite a un calcolatore.

programma biblioteca / library program Programma al quale possono accedere e che possono eseguire tutti gli utenti di un elaboratore.

programma di controllo / monitor Programma basico di sistema che assicura l'esecuzione ordinata di tutte le azioni di un calcolatore.

programma supervisore / executive system Programma basico di sistema che assicura l'esecuzione ordinata di tutte le azioni del calcolatore. Si usa anche il termine *monitor* (in inglese)

programmatore / programmer Specialista incaricato di scrivere un programma.

programmazione strutturata / structured programming La scrittura di un programma sotto forma modulare.

prova / testing Esecuzione di un programma con dati realistici per determinare l'esattezza della logica del programma.

pseudo codice / pseudo code Istruzioni espresse in un linguaggio per metà in lingua, per metà nel linguaggio dei calcolatori.

punto di implementazione / location Sinonimo di *parola* e *indirizzo*.

rappresentazione grafica / display Rappresentazione grafica di dati su un'unità d'uscita.

rappresentazione grafica / graphics Rappresentazione di dati sotto forma grafica.

realizzazione / implementation Applicazione di un programma o sistema.

redazione dei dati / editing Messa in forma e verifica dell'accuratezza dei dati.

registrazione / record Suddivisione logica di un file di dati. Una registrazione contiene l'insieme dei dati correlati ad una entità del file.

remote job entry / remote job entry (Rje) Inizio dell'esecuzione di un programma o insieme di programmi da una stazione a distanza. Il termine viene anche riferito alla trattazione di nastri off-site, alla lettura di schede, alla perforazione e stampa di tabulati mediante un'unità terminale Rje che può collegarsi, in genere a mezzo telefono, a un calcolatore.

rete / network Metodo di comunicazione con un calcolatore che utilizza di norma linee telefoniche. Il termine può riferirsi a funzioni e relazioni all'interno di un sistema di gestione di un progetto.

riconoscimento ottico di caratteri / optical charater recognition Scansione dei dati d'ingresso a mezzo o di una unità ottica che trasforma informazioni visive in impulsi elettronici.

rigenerare / refresh Atto mediante il quale i dati divengono contigui. Il termine viene usato anche per indicare l'eliminazione di spazio inutilizzato in un disco.

ripristinare / restore Restituzione di qualcosa al suo stato precedente . I dati possono essere restituiti da disco a nastro. I dati su nastro sostituiscono dati esistenti sul disco. Qualsiasi aggiornamento di dati su disco, ma non acquisito su nastro, va perduto.

risorsa / resource Qualsiasi parte del calcolatore.

risposta vocale / voice answer-back Risposta, a mezzo voce creata meccanicamente, data da un calcolatore ad una domanda sottoposta premendo i tasti di un telefono o tastiera sonora.

routine / routine Programma o sezione di un programma.

run / run Atto esecutivo delle istruzioni di un programma.

run-time system / run-time system Sistema che assolve funzioni, nel corso dell'esecuzione di un programma, alle quali il programmatore non ha bisogno di interessarsi quando scrive programmi con linguaggio ad alto livello quali il COBOL e FORTRAN.

scalare / scalar Variabile non appartenente ad un ordine.

scansione / scanning Scansione di dati, di cui un esempio è quella ottica.

scheda perforata / punched card Scheda di cartoncino nella quale le informazioni sono memorizzate da serie di perforazioni.

schermo / screen Schermo televisivo sul quale vengono rappresentati i dati.

screen entry / screen entry Immissione di dati in un calcolatore mediante l'impiego di un CRT, effettuata, in genere, con l'ausilio di una maschera di fondo.

scrivere / write Termine genericamente riferito alle istruzioni di un programma mediante le quali i dati vengono messi in una memoria o su un disco.

scrivere sopra / overwrite Sostituzione delle informazioni esistenti.

sequenza in linguaggio sorgente / source code Istruzioni di un programma sorgente

sistema / system Termine riferito sia alla configurazione dell'hardware di un sistema, sia a un insieme di programmi che complessivamente servono a risolvere un problema.

sistema decimale / decimal system Sistema di numerazione in base 10.

sistema di gestione della base dati / data base management system Sistema dotato di capacità gestionali e amministrative per il controllo della selezione delle registrazioni, degli aggiornamenti e dei rapporti da una base dati.

sistema di modellistica finanziaria / financial modeling system Programma che fornisce in uscita dati finanziari basati su calcoli di dati in entrata secondo un modello predeterminato. Di norma possono essere elaborati calcoli *e se*, dove l'uscita di un certo numero di elementi dei dati cambierà di valore, in dipendenza di una variazione del valore di uno o più elementi dei dati in entrata.

sistema operativo / operating system Programma basico di sistema che assicura l'ordinata esecuzione di tutte le azioni di un calcolatore.

software / software Termine usato per indicare il complesso dei programmi elaborati da un calcolatore.

software di sistema / system software Vedere *software*.

spazio del disco / disk space Spazio disponibile per la memorizzazione di dati su un disco.

stampante / printer Unità che stampa i dati di uscita di un calcolatore.

stampante pagina per pagina / page printer Unità stampante che stampa le pagine delle informazioni una alla volta.

stampante per linee / **line printer** Unità che stampa ad alta velocità, su moduli in continuo, le informazioni emesse dal calcolatore.

struttura / structure Configurazione di un programma. Termine usato anche per indicare l'interrelazione logica degli insiemi di dischi.

struttura d'archivio / file structure Configurazione di un archivio dati. Termine usato anche per indicare insiemi di

dischi considerati dal calcolatore per formare un'unità di immagazzinamento.

struttura logica / logical disk structure Più insiemi di dischi considerati logicamente come unità di immagazzinamento da un calcolatore.

sviluppo / development Operazione mediante la quale si attuano l'analisi, il design e il programma per la soluzione di un problema.

sviluppo di sistemi / system development Vedere *sviluppo*.

tabulato / listing Prospetto di un programma o di un archivio dati ottenuto da una unità stampante.

terminale / terminal Unità a tastiera che può essere collegata ad un computer allo scopo di immettere e ricevere dati.

terminale di calcolatore / computer terminal Macchina a tastiera che in collegamento con un calcolatore può essere usata per immettere e per ricevere dati.

terminale intelligente / intelligent terminal Terminale in grado di eseguire elaborazioni in entrata e uscita, dotato di capacità logiche autonome.

testina / head Testina che legge e scrive dati da e su disco.

testa di lettura-scrittura / read-write head Testina che deposita i dati su un disco sotto forma di particelle magnetiche. La stessa unità è anche in condizione di leggere i dati da un disco e depositarli in una memoria.

time sharing / time sharing Utilizzazione di un calcolatore da località distanti, da parte di uno o più utenti contemporaneamente.

throughput / throughput Velocità e capacità di un calcolatore, misurate dal tempo richiesto per produrre i risultati desiderati.

traduzione / translation Trasformazione di un codice scritto in un dato linguaggio in istruzioni di un altro linguaggio.

trasferimento automatico di fondi / automatic fund transfer Collegamento diretto fra elaboratori, che tratta il trasferimento di denaro da un conto bancario ad un altro, senza il ricorso ad operazioni d'ufficio tradizionali.

trasmissione / transmission Invio e ricezione di dati a e da un calcolatore, su linee telefoniche.

traccia / track Unità di immagazzinamento su disco.

tubo a raggi catodici / cathode ray tube (CRT) Terminale di visualizzazione a tubo catodico di un computer. Schermo televisivo per mezzo del quale si accede a programmi e dati del calcolatore.

ubicazione della memoria / memory location Sinonimo di *parola*, *cassetta per lettere*, *indirizzo*, e così via.

unità a nastro / tape drive Unità che legge e scrive nastri magnetici.

unità centrale di elaborazione / central processing unit Componente del calcolatore dove vengono eseguiti tutti i calcoli.

unità di immagazzinamento / storage device Mezzo fisico per immagazzinare dati, come un disco o una bobina per nastri.

unità disco / disk drive Macchina contenente l'insieme di dischi. L'unità ha testine di lettura/scrittura che possono depositare i dati sul disco e riportarli dallo stesso.

unità periferica / peripheral device Unità esterna, come una stampante o una unità a disco, non indispensabile alle operazioni basiche di un calcolatore.

uscita / output Termine con il quale si indica il trasferimento dei dati da un calcolatore a nastri, stampanti, rappresentazioni grafiche, plotters, terminali.

valore / value Contenuto dell'indirizzo di una memoria.

velocità / speed Termine riferito a: – velocità con la quale si può accedere ai dati su disco; – velocità con la quale i dati possono essere depositati su disco; – velocità con la quale la CPU può trattare le istruzioni; – durata del ciclo della memoria.

verifica dei dati / data verification Controllo della esattezza dei dati.

verifica delle schede / card verification Processo che consente di accertare l'esattezza dei dati codificati sulle schede perforate.

L'hardware

Un calcolatore, per definirsi tale, deve essere costituito da determinate apparecchiature. In questo capitolo esamineremo queste apparecchiature, vale a dire la componente "hardware" del calcolatore.

L'unità centrale per l'elaborazione

Il primo elemento è la CPU (Central processing unit). Si tratta del cuore della macchina, dal momento che tutti i calcoli vengono effettuati qui. La CPU può essere piccola quanto il tradizionale macinino del caffè – qualche volta anche più piccola – oppure può essere grande quanto una coppia di armadi guardaroba. Ma, qualunque sia la sua dimensione, è il cuore del sistema e di norma anche la parte più costosa del calcolatore.

Per coloro che sono stati all'interno di locali dove sono installati dei calcolatori vale ricordare che tutte le lampadine lampeggianti sulla CPU e su altri hardware non hanno alcun significato per nessuno, quando la macchina funziona normalmente; esse sono utili solo se la macchina sbaglia. In questo caso le lampadine sono in grado di indicare l'esistenza di un problema nell'hardware. Disgraziatamente anche le lampadine possono sbagliare e se ciò accade, l'indicazione del guasto diventa inesatta. Per la verifica delle lampadine esistono appositi programmi, ma la loro sostituzione è costosa; di qui lo sviluppo di nuovi strumenti per la rilevazione guasti. In futuro sull'hardware vi sarà un minor numero di lampadine lampeggianti, ed i locali in cui verranno installati i calcolatori saranno completamente senza illuminazione. In

essi si troverà, invece, una serie di apparati, in apparenza inattivi, disposti tutt'intorno alle pareti. L'unico evidente segno di *vita* sarà il mormorio delle apparecchiature per l'aria condizionata.

La memoria

La seconda parte integrante di un calcolatore è denominata memoria. La memoria è la parte della macchina dove risiedono, quando l'elaboratore funziona, programmi e dati. Ciò significa che quando il calcolatore esegue un programma, lo stesso programma, le informazioni e i dati di cui ha bisogno, sono situati in questa memoria. Vi è un'azione reciproca continua fra CPU e memoria e quando vengono eseguite le istruzioni per un programma, viene coinvolta la CPU.

Supponiamo di voler eseguire la somma di due numeri. La memoria ha i due addendi dell'addizione immagazzinati in qualche parte. La CPU sommerà le due cifre e depositerà il risultato dell'addizione nella memoria. Il programmatore incaricato di compiere questa operazione non ha bisogno di sapere nulla della interazione fra CPU e memoria. Il suo unico compito è dare istruzioni al calcolatore, per mezzo di un'espressione scritta in linguaggio programmato, per effettuare l'addizione. Le istruzioni del linguaggio programmato non hanno niente di magico. Per ottenere che il calcolatore svolga un'addizione, basta questa semplicissima indicazione:

poniamo A=B+C

Si dà il caso che si tratti di un'espressione accettabile dal linguaggio di programmazione denominato Basic. Quando il programmatore scrive questa frase, in effetti dà istruzioni al calcolatore di addizionare i valori che al momento risiedono nella posizione B e C della memoria e di depositare il risultato di questa addizione nella posizione A della memoria.

Ciò sta a significare che la memoria è divisa in tante *posizioni*. Ciascuna posizione, o parola, nella memoria può contenere un *valore*. Potete paragonare la memoria di un calcolatore ad una serie di cassette per la corrispondenza di un condominio. Nel loro insieme esse costituiscono un punto

di recapito della posta per l'intero condominio, mentre ciascuna cassetta ha il proprio indirizzo. Il postino deposita in ogni cassetta le lettere che formano il *contenuto* o *valore* della cassetta. Esattamente allo stesso modo ci si indirizza alla memoria. La serie di tutte le cassette per corrispondenza è la memoria stessa; le cassette individuali vengono chiamate indirizzi, o nuclei di memorie o parole. I contenuti di una *cassetta per la corrispondenza* di un calcolatore vengono chiamati valore, o contenuto degli indirizzi di memoria.

Il termine nucleo di memoria ebbe origine al tempo in cui le memorie venivano realizzate con centinaia di migliaia di fili incrociati. Dovunque questi fili si incrociavano venivano legati con un anello minuscolo denominato *nucleo*. Oggi, le memorie utilizzano una tecnologia più avanzata e i fili sono scomparsi. Tuttavia il termine *nucleo* è rimasto, ed è usato per indicare l'indirizzo.

Nonostante *nucleo* e *parola* siano usati indiferrentemente quando ci si riferisce a un indirizzo, in realtà non sono sinonimi. Allo scopo di formare un indirizzo della memoria, o una parola della memoria, molti *nuclei* si considerano *appartenenti alla stessa famiglia*. Alcuni calcolatori mettono 16 *nuclei* assieme per formare una parola; altri ne impiegano 32, altri ancora ne impiegano 36 o più. Così possiamo raffigurarci le parole della memoria costituite da un certo numero di nuclei, ciascuno dei quali, noto come *bit* (che sta a significare *cifra binaria*), può assumere uno dei due *valori*: *on* o *off*. Un bit è *on* quando è presente un impulso elettrico ed è *off* quando tale carica elettrica non esiste. Quando un bit è *on* si rappresenta come un 1, quando invece è *off* viene usato lo 0.

Graficamente guardiamo ai bit di una parola da destra a sinistra. Di seguito è indicata una configurazione binaria della parola di memoria che contiene solo quattro bit:

Numero del bit	4	3	2	1
Rappresentazione di tutti i quattro bit *on*:	1	1	1	1
Rappresentazione di tutti i quattro bit *off*:	0	0	0	0

Vediamo quali valori numerici una parola della memoria può contenere, valendoci della teoria *on-off* (ovvero 1-0). Supponiamo che una certa memoria abbia parole di quattro bit ciascuna. Si applica la regola seguente: se solo il bit 1 è *on*,

il valore della parola è 1; se solo il bit 2 è *on*, il valore della parola è 2; se solo il bit 3 è *on*, il valore della parola è 4; se solo il bit 4 è *on*, il valore della parola è 8. Secondo questa regola possiamo formare tutto il complesso dei numeri da 0 a 15 nell'ambito di questa parola a quattro bit. Facciamo una prova: quando tutti i bit sono *off*, il valore che la parola rappresenta è 0; quando i bit 1 e 2 sono *on* il valore della parola è 1+2=3, e via di seguito. Questa *regola*, secondo la quale ciascun bit consecutivo rappresenta un valore doppio di quello immediatamente precedente (1, 2, 4, 8, 16, 32 e così via) viene definita numerazione in base 2, e rappresenta il principio d tutti i calcoli dei valori all'interno del calcolatore. La tabella che segue riassume la nostra discussione sui valori.

Ora sono anche evidenti le limitazioni della grandezza dei valori che possono essere immagazzinati in ciascuna parola della memoria di un calcolatore. Non sarà possibile immagazzinare un numero maggiore di 15 in una parola che ha soltanto quattro bit.

SOMMARIO DEI VALORI BINARI

Valore se il bit è *on*	8 4 2 1
Numero del bit	: 4 3 2 1
Zero bit *on*	: 0 0 0 0 (valore =0)
Un bit *on*	: 0 0 0 1 (valore =1)
Due bit *on*	: 0 0 1 1 (valore =1+2=3)
Tre bit *on*	: 0 1 1 1 (valore =1+2+4=7)
Quattro bit *on*	1 1 1 1 (valore =1+2+4+8=15)
Il valore 5 sarà	: 0 1 0 1 (valore =1+4=5)
Il valore 9 sarà	1 0 0 1 (valore =1+8=9)
Il valore 13 sarà	1 1 0 1 (valore =1+4+8=13)
e così di seguito.	

Per immagazzinare valori più grandi i calcolatori hanno bisogno di dimensioni maggiori della parola: 32 bit o più per parola sono comuni nei calcolatori del giorno d'oggi. Il nu-

mero massimo che può essere immagazzinato in una parola lunga 32 bit è 2.147.483.647, che effettivamente occupa solo 31 bit della parola.

Per indicare se il valore è positivo o negativo, è necessario un bit della parola.

Il valore dei bit (on-off) di una parola viene spesso espresso con la notazione ottale (base 8) o esadecimale (base 16) anziché con la notazione decimale (base 10). Qui di seguito viene riportato un confronto e la spiegazione, solo per amore di completezza. Se però volete saltare la lettura di questo breve capitolo tecnico ritenetevi liberi di farlo. Non avete alcun bisogno di sapere niente di questo problema per comprendere il modo di operare dei calcolatori. Intendo solo assicurare che la gente che si esprime secondo le notazioni esadecimale o in base 8 non vi faccia sentire in stato d'inferiorità. Esprimere il valore di una parola in qualsiasi notazione non modifica affatto il valore della parola. Il tutto è molto simile al cambio delle lire in moneta straniera quando comperate una stecca di sigarette su un aereo nel corso di un viaggio internazionale: qualunque valuta usiate per pagare la merce, ed in qualunque modo si calcoli la conversione, il valore di ciò che ricevete rimane lo stesso. Addirittura, quando il *fattore confusione* comincia ad intromettersi nella conversazione con gli esperti di calcolatori, chiedete loro di esprimere i valori in *base 45*. Osservate poi la loro incredulità e la loro ammirazione per la vostra ovvia, superiore conoscenza.

RAFFRONTO FRA VALORI BINARI E ALTRI VALORI

Valore binario	Valore decimale	Valore esadecimale	Valore ottale
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7

Valore binario	Valore decimale	Valore esadecimale	Valore ottale
1000	8	8	
1001	9	9	
1010	10	A	
1011	11	B	
1100	12	C	
1101	13	D	
1110	14	E	
1111	15	F	

La tabella di raffronto vi chiarirà come sono espressi i valori nelle notazioni di uso più frequente. Anzitutto vi esporrò la notazione binaria (bit *on* o *off*, da destra a sinistra), poi l'equivalente decimale, infine la esadecimale (abbreviazione *hex* o base 16), seguita dalla notazione ottale (base 8).

Nella prima riga della tabella si vede che tutti i bit (da 1 a 4 da destra a sinistra) sono *off*, perché sono tutti rappresentati dallo 0. Come abbiamo visto in precedenza, il valore decimale di questo modello di bit è 0. Per semplificare le cose nelle altre notazioni, il valore di questo modello di bit è 0 anche nelle notazioni esadecimale e ottale. Indico gli stessi valori per tutte tre le notazioni per tutto il modello di bit 0111, che, ricorderete, espone il valore decimale 7 (da destra a sinistra: 1+2+4=7).

La notazione ottale, in base 8, ora ha completato la sua *base*. La notazione ottale raggruppa perciò set di tre bit insieme, rappresentati da destra a sinistra nella tabella, per esprimere il valore di una parola nella memoria.

La scala dei valori esadecimali continua, ma quando giungiamo ad una notazione binaria di 1010, per un valore decimale di 2+8=10, la notazione esadecimale di questo valore diventa la lettera A. La tabella prosegue fino a che anche la notazione esadecimale ha completato la sua *base*, quando i bit sono tutti *on* nel modello quattro bit e il valore, espresso nella notazione decimale, è diventato 1+2+4+8=15. La notazione esadecimale raggruppa insieme quattro bit, da destra a sinistra, per esprimere il valore di una parola della memoria.

In base a ciò, riportiamo altri esercizi. Supponiamo di avere una parola di 16 bit. Nella notazione binaria il valore della parola viene espresso così:

0 1 1 1 0 1 1 1 0 0 1 0 1 0 1 0

Diamo un'occhiata al valore decimale di questa parola:

```
  1
  6 8 4 1
  3 1 0 0 5 2
  8 9 9 2 1 5 3
  4 2 6 4 2 6 2 8 2
0 1 1 10 1 1 10 0
```

La notazione decimale, da destra a sinistra, è:
2+8+32+256+512+1.024+4.096+8.192+16.384=30.506.

La notazione esadecimale raggruppa insieme quattro bit, da destra a sinistra:

0 1 1 1 0 1 1 1 0 0 1 0 1 0 1 0

Cominciando, in notazione esadecimale, sul lato destro della parola e operando verso sinistra:

bit 1, 2, 3, 4 ammontano a 0+2+0+8=valore hex A
bit 5, 6, 7, 8 ammontano a 0+2+0+0=valore hex 2
bit 9, 10, 11, 12 ammontano a 1+2+4+0=valore hex 7
bit 13, 14, 15, 16 ammontano a 1+2+4+0=valore hex 7

Ora possiamo scrivere il suddetto valore binario di 16 bit come esadecimale:

772 A

Di seguito è riportata la stessa notazione binaria tradotta in valore ottale. Ancora da destra a sinistra, tre bit sono raggruppati insieme:

0 1 1 1 0 1 1 1 0 0 1 0 1 0 1 0

bit 1,2 e 3 ammontano a 0+2+0=valore ottale 2
bit 4, 5 e 6 ammontano a 1+0+4=valore ottale 5

bit 7, 8 e 9 ammontano a 0+0+4=valore ottale 4
bit 10, 11 e 12 ammontano a 1+2+0=valore ottale 3
bit 13, 14 e 15 ammontano a 1+2+4=valore ottale 7
bit 16 =valore ottale 0

Lo stesso valore binario ora può essere scritto in notazione ottale e cioè:

73452

Ora i termini *ottale*, *esadecimale*, cioè *base 8* e *base 16* non vi spaventeranno più. Come ho affermato prima, potete dimenticare tutte queste varie notazioni dei valori. Ricordate soltanto le *parole* in cui si divide la memoria e i bit binari (*on-off*) che insieme formano il valore contenuto in ciascuna parola.

Un altro argomento degli uomini Edp, dopo le dimensioni della parola, sarà la capacità della memoria stessa. La capacità di memoria denota il numero delle parole di memoria disponibili in un particolare sistema. La capacità di memoria è in funzione del numero delle parole che contiene. Di norma le dimensioni sono misurate in multipli di 1.024 parole. Un'unità di 1.024 parole viene definita *k*. Alcuni produttori, Ibm inclusa, calcolano la dimensione delle memorie in *byte*, anziché in parole, dove un *byte* consiste in otto bit.

La dimensione può anche essere riferita al numero di bit contenuti in ogni parola, così non sorprendetevi quando sentite parlare di una "memoria 512 k 36 bit". Tutto ciò sta a significare che ogni parola di questa particolare memoria ha 36 bit e che state parlando di una memoria che ha 512 x 1.024=524.288 parole, o indirizzi, o cassette per lettere.

Il magazzino permanente

La terza componente di un calcolatore è l'apparecchiatura che va sotto il nome di *magazzino permanente*. Si tratta della parte del calcolatore dove vengono conservate le informazioni quando non sono nella memoria. Quando si creano o si aggiornano dati e programmi, le informazioni relative vengono registrate sul magazzino permanente. Gli apparati odierni di magazzinaggio permanente sono macchine che assomigliano a giradischi futuristici. Da solo l'apparato di immagazzinamento, denominato *disk pack*, cioè insieme di

dischi, consiste in un certo numero di dischi simili agli LP. Questi sono montati su di un perno, separati da un piccolo spazio per consentire il movimento delle *testine di lettura e scrittura* simili al braccio dei giradischi. Oggi, uno di questi congegni può contenere fino a 800 milioni di caratteri di informazione. Per darvi un'idea della quantità dei caratteri, cercate d'immaginare un libro di 300.000 pagine. Si tratta di un libro macroscopico, eppure contiene soltanto le informazioni che risiedono in appena un *disk pack*.

A differenza dei dischi del vostro stereo, i dischi di un'unità di magazzino permanente di un calcolatore possono essere cancellati e riscritti con nuove informazioni. Inoltre i programmi e i dati che sono immagazzinati nei *disk pack* sono *accessibili casualmente*, e ciò significa che, quando un programmatore dà istruzioni ad un calcolatore per l'individuazione di certe informazioni su disco, il calcolatore è in grado di trovarle quasi direttamente, senza dover prima esaminare tutte le informazioni precedenti.

Rendiamoci conto un po' più da vicino dell'immagazzinamento di dati su disco. Come ho già accennato ciascun disco ha una *testina* che legge e scrive dati da e su disco. Queste informazioni vengono immagazzinate e, quindi, memorizzate, in sezioni del disco denominate da certi produttori *track* e da altri *block*. Qualunque sia il termine usato, i dischi sono divisi in segmenti. Ciascun segmento contiene un certo numero di parole, come nella memoria, e ciascuna parola è a sua volta suddivisa in bit.

Memoria e disco usano le stesse regole di nomenclatura, ma il modo di magazzinaggio dei dati presenta due differenze fondamentali. La prima sta nel fatto che i dati *immagazzinati* nella memoria sono rappresentati da impulsi elettronici e pertanto se spegnete il calcolatore o se si verifica qualche difetto di funzionamento tutti i dati presenti in quel momento nella memoria (nei nuovi tipi di memoria a semiconduttori) vanno perduti. Il disco, al contrario, immagazzina i dati magneticamente. Ciò significa che i dati sul disco sono permanenti, indipendentemente dal fatto che il calcolatore sia acceso o spento ed esisteranno fino a quando non saranno volutamente cancellati e sostituiti.

La seconda differenza sta nel fatto che la memoria distingue fra dati numerici ed alfanumerici. I dati numerici sono, di fatto, numeri con cui si possono eseguire operazioni arit-

metiche. I dati alfanumerici sono, invece, lettere dell'alfabeto, che possono anche apparire per rappresentare numeri. La cosa confonde un po', meglio fare un esempio: il carattere 9 può essere rappresentato come numero, ma anche come carattere alfanumerico. Se è rappresentato come carattere alfanumerico si possono veramente eseguire operazioni aritmetiche ma non può essere stampato come un numero. Il calcolatore riceve istruzioni dal programmatore per memorizzare un carattere numerico o alfanumerico ed il magazzino del disco non fa distinzioni: tutto è immagazzinato come se si trattasse di dati numerici. È competenza del programmatore ricordare se una parola del magazzino del disco rappresenta un numero ovvero una sequenza alfanumerica. In pratica non è molto difficile conservare traccia di ciò: il programma usato per scrivere i dati sul disco specificherà quali sono le parole numeriche e quali sono le alfanumeriche, così da permettere un controllo del programma con estrema facilità.

Le parole a memoria su disco sono spesso suddivise in *byte*. Byte è un termine non scientifico che denota la quantità di bit utilizzati per formare un carattere. Alcune industrie di elettronica impiegano byte di sette bit, altre byte di sei bit, altre ancora byte di otto bit. Ma in generale, quando si parla di byte, ci si riferisce ai caratteri. L'Ibm ha sempre definito il *byte* composto di otto bit, cosicché per gli utenti Ibm e di altri calcolatori simili, il byte risulta un termine definito molto chiaramente.

Come abbiamo visto prima un *disk pack* può contenere milioni di byte, o di caratteri in memoria. Finora questo libro contiene circa 38.000 caratteri (contando anche gli spazi fra le parole poiché, dopo tutto, anche uno spazio è un carattere) o all'incirca 1/21052 di un'unità magazzino di 800 milioni di caratteri. Tuttavia molte basi di dati richiedono più ampie disponibilità di memoria rispetto a quelle fornite da una sola unità. Se si verifica un caso del genere si riunisce un certo numero di dischi nella logical disk structure, *struttura logica a dischi*. Ciò significa che il calcolatore considera i *disk pack* separati come una *unità* e quindi risulta possibile leggere e scrivere i dati come se l'intera struttura fosse un enorme insieme di dischi.

Entrata e uscita

Ed eccoci qui. Ora disponiamo di un calcolatore con un'unità centrale di elaborazione, una memoria ed un magazzino permanente, tutti operanti insieme per darci quei rapidi risultati ai quali siamo abituati. Il magazzino permanente interagisce con la memoria, che a sua volta interagisce con l'unità centrale di elaborazione. Rendiamoci conto, con un esempio, di queste interazioni. Supponiamo sia stato scritto un programma che prevede l'addizione di due numeri. Questo programma è stato memorizzato nel magazzino permanente. Ora vogliamo eseguire (svolgere) questo programma e depositare nuovamente il risultato dell'addizione nel magazzino permanente, cosicché più avanti nel tempo (domani, il mese prossimo, o fra qualche anno) potremo utilizzare il risultato per altri calcoli e situazioni. Ecco ciò che accade nel nostro calcolatore, tutto sotto controllo del programma in corso di esecuzione (il calcolatore da solo non prende alcuna decisione):

1 – Il programmatore dice all'unità centrale di elaborazione di eseguire il programma.
2 – L'unità centrale di elaborazione trova il programma nel magazzino permanente.
3 – Il programma viene riprodotto entro la memoria dal magazzino permanente.
4 – L'unità centrale di elaborazione calcola il risultato e lo dice alla memoria.
5 – Lo spazio del magazzino permanente viene allocato su disco.
6 – Il programmatore viene informato che il programma è finito.

Si tratta di pochi eventi, il tutto però viene fatto in un tempo dell'ordine di microsecondi.

Oggi se si affiancano un'unità centrale di elaborazione, una memoria e poche unità magazzino il risultato non è molto impressionante. Lo è stato a suo tempo. I calcolatori nei primi tempi del loro sviluppo (1940-1960) occupavano tranquillamente due o tre piani di un edificio. Molto spazio era usato dall'impianto dell'aria condizionata. I calcolatori impiegavano centinaia, talvolta migliaia di valvole termoioniche. I sistemi di raffreddamento necessari per tenere quelle valvole produttrici di calore ad una temperatura operativa

confortevole erano così grandi che le esigenze elettriche di una piccola città spesso avrebbero potuto esser soddisfatte dalla potenza richiesta per raffreddare un solo calcolatore. L'invenzione del transistor e altri progressi tecnologici hanno cambiato tutto ciò: i calcolatori sono diventati molto più piccoli, molto più freddi e nel contempo molto più potenti. Oggi alcuni calcolatori da tavolo hanno prestazioni superiori a quelle di un calcolatore a tre piani di 20 anni fa.

Gli unici componenti che appaiono ancora impressionanti e rendono interessante una visita nei locali dove è situato un calcolatore sono quelli periferici. Le unità periferiche sono le macchine che permettono l'interazione umana con il calcolatore. Unità a disco, macchine lettrici di schede perforate, perforatrici, terminali con tastiera, video, stampanti e plotter costituiscono le apparecchiature terminali. Nelle installazioni dove le schede perforate sono lo strumento principale per l'immissione di dati nel calcolatore, troverete anche classificatori di schede, ordinatori e apparecchiature per la verifica delle schede.

I programmi televisivi e gli annunci economici ricorrono spesso a unità a disco rotanti per descrivere i calcolatori. Infatti la bobina del nastro magnetico è diventata il simbolo dell'industria dei calcolatori: quando vediamo unità a nastro pensiamo immediatamente ai calcolatori, anche se esse hanno poco a che fare con il lavorìo di base degli stessi. I nastri sono mezzi per l'immagazzinamento, impiegati per conservare dati e programmi e per gli stessi scopi dei *disk pack* già descritti. La grande differenza fra dischi e nastri consiste nel fatto che l'accesso ai nastri è esclusivamente sequenziale, non casuale. Ciò vuol dire che se il calcolatore ha bisogno di informazioni memorizzate su nastro, deve leggere sullo stesso tutte le informazioni che precedono il programma cercato. Si tratta di un processo relativamente lento. La lettura su un nastro intero può richiedere fino a 30 minuti, mentre su un disco le informazioni si trovano, normalmente, in una sola frazione di secondo.

La popolarità dei nastri deriva dal fatto che grazie ad essi risulta possibile ordinare e scambiare dati provenienti da due o più fondi. La combinazione dei dati si svolge nel seguente modo. Supponiamo di avere due insiemi di dati, ognuno di questi immagazzinato su nastri separati con i dati di ciascun nastro già ordinati. Ciò significa che se abbiamo

dati contenenti i nominativi di determinate persone, questi nomi saranno in sequenza alfabetica logica come in un elenco telefonico. Se volessimo, ad esempio, combinare informazioni sugli abbonati di due periodici e farle arrivare ad un elenco di indirizzi postali congiunto dovremmo combinare i due nastri in uno (ancora ordinato secondo sequenza alfabetica) usandone un terzo, chiamato *pulito*, cioè senza alcuna informazione all'inizio del processo di fusione. Il programma del calcolatore *leggerà* un nome dal primo nastro, poi *leggerà* un nome dal secondo nastro, quindi stabilirà il nome che viene prima lo scriverà sul nastro pulito. *Leggere* è il termine riferito alla sensibilizzazione delle particelle magnetiche sul nastro, quindi al deposito nella memoria delle configurazioni binarie risultanti; *scrivere* significa trasferire le parole della memoria in configurazioni binarie e depositarli come particelle magnetiche sul nastro.

Dopo aver scritto un nome sul terzo nastro, il programma darà ordini al calcolatore di ricordare il nome non ancora scritto sul terzo nastro, perché, naturalmente, avremo bisogno di questo nome più tardi. Il nome successivo viene letto dal nastro che ci ha dato il nome che abbiamo appena scritto sul terzo nastro; adesso il programma confronta questo nuovo nome con quello ancora immagazzinato nella memoria. Di nuovo, qualunque nome che alfabeticamente viene prima di un altro, viene trascritto. Quando i due nastri giungono alla fine, e dopo che il programma *cancella la sua memoria* scrivendo sul terzo nastro l'ultimo nome che ha ancora immagazzinato, il programma è finito e tutti i nomi di entrambi i nastri ora sono combinati in sequenza, ordinata alfabeticamente, sul nastro pulito che adesso diventa il nastro *master* (cioè quello principale, utilizzabile per la duplicazione).

Oggigiorno operazioni di classificazione e fusione vengono spesso effettuate su insiemi di dischi, ma i nastri continueranno a giocare un proprio ruolo sulla scena dei calcolatori. I nastri costano meno dei *disk pack* e sono più facili da trattare, inoltre sono spesso impiegati per immagazzinare programmi e dati *fuori linea*. Ciò significa che le informazioni al momento non necessarie su disk pack vengono scritte su nastro magnetico e cancellate dal disco. Questo processo libera spazio utile dal disco mentre trattiene su nastro le informazioni, in forma leggibile per un calcolatore. Se, e quando queste informazioni sono di nuovo necessarie, il nastro può essere

letto dal calcolatore e le informazioni trasferite di nuovo su disco.

I nastri continueranno ad essere usati ancora per *fare la fotografia*, a intervalli regolari, di tutte le informazioni presenti in un insieme di dischi. Molte installazioni eseguono tale operazione con frequenza giornaliera o settimanale. *Fare la fotografia* significa che tutte le informazioni immagazzinate nei dischi vengono scritte, per sicurezza, su nastro magnetico. Nel caso succeda qualcosa per cui il disco diventa *illeggibile* per il calcolatore, questi nastri *sicuri* vengono usati per ripristinare le informazioni corrette dal momento d'applicazione della procedura di copiatura. Problemi che si traducono nella perdita di dati (denominata come *distruzione del disco*) oggigiorno sono rari, ma possono verificarsi e le procedure di sicurezza sono pertanto essenziali per assicurare che la perdita di un disco non comporti la perdita di tutti i dati inseriti sullo stesso. Se la procedura da disco a nastro viene seguita giornalmente, il peggio che può capitare in caso di distruzione di un disco equivale alla perdita dell'aggiornamento giornaliero dei dati, ma è di gran lunga meglio perdere il lavoro di un giorno solo che perdere tutti i dati, evento che richiederebbe mesi o anni per ripristinare la situazione.

Un altro metodo per immettere e leggere dati in un calcolatore è quello delle schede perforate. È probabile che abbiate sentito parlare di *perforazione*. Gli addetti alla perforazione trascrivono dati da documenti scritti su schede che possono essere lette da macchine lettrici. Le fatture che ricevete ogni mese sono spesso sotto forma di schede perforate. Il vostro nome, indirizzo, numero di conto corrente e saldo debitore sono tutti elementi perforati sulla scheda. Quando rispedite la scheda con allegato il vostro assegno, un operatore perfora la scheda per l'importo pagato, la scheda viene letta e i dati relativi al pagamento entrano a far parte della vostra registrazione nel magazzino permanente del calcolatore. Spesso sulla scheda sono stampate le avvertenze di non spillare, piegare o manomettere la scheda stessa. La ragione sta nel fatto che se la vostra scheda viene *deformata*, non può esser letta dalle apparecchiature per la lettura automatica. Nel passato i programmi dovevano essere sempre perforati su schede prima che il calcolatore potesse eseguirli; oggi, anche se in molte installazioni vengono impiegate le schede per questo scopo, i programmi possono essere dati ai calcola-

tori direttamente, tramite terminali con tastiera. La scheda perforata è una delle forme più vecchie di *comunicazione* con un calcolatore ed è ancora molto impiegata.

Un altro sistema di lettura dati del calcolatore è quello di lettere e numeri codificati magneticamente, stampati su documenti. I vostri assegni bancari, ad esempio, sono codificati magneticamente. Sui libretti degli assegni che vi vengono rilasciati, il numero del vostro conto corrente e quello della banca sono codificati; quando pagate una fattura con un assegno, l'impiegato della banca codifica la somma sull'assegno che viene letto da una apparecchiatura che traduce le particelle magnetiche in forma leggibile. Spesso si possono rintracciare errori bancari seguendo la codificazione errata dell'ammontare della somma.

Un altro sistema ancora, mediante il quale il calcolatore può leggere informazioni, è denominato riconoscimento ottico di caratteri *(Ocr)*. Vi saranno senz'altro note le forme della scansione ottica. Esse sono utilizzate spesso in rilevazioni di tutti i generi e come moduli d'ordine per vari prodotti. Vi viene chiesto di riempire un quadrato o un rettangolo, o di scrivere una X in una determinata casella, ovvero di formare dei numeri in modo piuttosto strano. Per fare queste operazioni vi si avvisa di *usare una matita n. 2*. La ragione per cui va usato tale tipo di matita, sta nel fatto che è stato accertato che il grado di morbidezza della mina consente la massima leggibilità al lettore ottico di caratteri. Le apparecchiature Ocr sono ancora allo stadio della prima infanzia, infatti possono essere lette solo fonti molto stilizzate; non la vostra calligrafia. Il vostro modulo, una volta ultimato, viene letto da un'apparecchiatura che trasforma le caselle riempite in impulsi che possono essere compresi dal calcolatore e, ancora una volta, la vostra registrazione viene aggiornata nel magazzino permanente del calcolatore.

Recenti sviluppi nel campo del riconoscimento ottico dei caratteri hanno portato alla produzione di un'apparecchiatura che può *imparare* le fonti nel momento in cui le legge. È in grado di ricordare le caratteristiche particolari di un insieme di lettere in poche prove, nel corso delle quali chiede all'utente di identificare i caratteri che non *capisce*, è in condizione di leggere documenti e perfino interi volumi. Ora è anche possibile trasformare la stampa tradizionale in caratteri braille o in una registrazione.

I calcolatori possono fornire informazioni mediante apparecchiature a scheda perforata. Le perforazioni sulle schede compiono la funzione opposta delle macchine lettrici di schede. Anziché leggere informazioni su un calcolatore interpretando i fori dalle schede, le perforatrici creano i fori sulle schede.

Abbiamo visto ora alcuni metodi per introdurre informazioni nel magazzino di un calcolatore impiegando una macchina per la lettura delle schede o una per il riconoscimento dei caratteri e per leggere informazioni da un calcolatore con una scheda perforata. L'immissione di informazioni in un calcolatore si definisce *entrata*, la ricezione di informazioni da un calcolatore di definisce *uscita*. In parole povere l'entrata si riferisce a dati che vanno nella memoria o magazzino permanente di un calcolatore, l'uscita si riferisce alle informazioni che il calcolatore trasferisce da un magazzino o da una memoria a qualche apparato di uscita, quali una stampante, un display, dei nastri magnetici e li rende disponibili per l'utente.

Se intendiamo trasferire dati da un calcolatore ad un altro, un modo è quello di avere le schede perforate del primo calcolatore con i dati da trasferire, per poi leggerle nel magazzino permanente dell'altro calcolatore mediante la sua macchina lettrice di schede. Esistono altri modi più rapidi per trasferire dei dati dalla macchina A alla macchina B. Attualmente è possibile realizzare anche collegamenti diretti fra calcolatori, *da elaboratore centrale a elaboratore centrale*, come si dice abitualmente.

Anche le installazioni che impiegano schede per l'entrata e l'uscita ricorrono ad ordinatrici. Si tratta di apparecchiature meccaniche che selezionano le schede secondo un ordine predeterminato. La selezione viene effettuata per mezzo di spazzolini metallici o fasci di luce che esplorano determinate colonne delle schede. Le schede vengono poi imbucate in distributori automatici in ordine selezionato. Nel passato questo metodo era spesso il migliore per avere dati in un certo ordine predeterminato. Con lo sviluppo dei programmi selezionatori generalizzati, la selezione viene spesso effettuata all'interno del calcolatore stesso. Ciò è molto più veloce ed anche maggiormente affidabile. Le schede tendono a bloccarsi nella macchina che cerca di selezionarle, specie se il grado di umidità non è entro certi valori accettabili.

Un altro metodo ancora per l'immissione di informazioni in un calcolatore è quello dei terminali con tastiera. Questi terminali sono collegati a un calcolatore sia direttamente (*hard-wired*) sia tramite collegamento telefonico. Informazioni dattiloscritte sulle tastiere diventano parte dei dati immagazzinati nella macchina. Il vantaggio di questo metodo consiste nel fatto che il programmatore non deve aspettare che la lettrice delle schede si liberi prima di affidare il programma al calcolatore e provarlo. Questo metodo di comunicazione con i calcolatori sta diffondendosi sempre più. Spesso, questi terminali con tastiera sono collegati a macchine a banda o a cassetta, così da far originare direttamente i dati in banda o cassetta senza bisogno di collegare il terminale con tastiera al calcolatore. Poi, quando tutti i dati sono *fuori linea*, il calcolatore è in grado di leggerti ad una velocità più elevata di quella realizzabile da una dattilografa. Se la macchina è collegata a un calcolatore mediante linea telefonica la tipica velocità di lettura di questa apparecchiatura è dell'ordine di 30-120 caratteri al secondo. Se il collegamento con il calcolatore è diretto (*hard-wired*) l'interazione, con l'uso di un terminale video, può spesso essere accelerata fino a 960 caratteri al secondo.

Queste interfaccia con tastiera possono anche essere impiegate per fornire dati dal calcolatore all'utente. Perciò, a livello di impiego operativo a bassa velocità, i *terminali*, sono spesso sufficienti per comunicare con un calcolatore, senza dover ricorrere per l'ingresso a lettrici di schede e per l'uscita a stampanti ad alta velocità.

Alla stessa categoria di apparati per le comunicazioni appartengono i terminali Crt (a raggi catodici), che assolvono le stesse funzioni dei terminali a tastiera ma, anziché fornire copie stampate di tutte le entrate e uscite proiettano le informazioni su di uno schermo televisivo. I vantaggi del terminale a raggi catodici consistono principalmente nell'assenza di rumore con cui opera e nella maggiore velocità con cui i dati compaiono sullo schermo. Naturalmente non s'impiega carta, ma l'uso di uno schermo televisivo come interfaccia con un calcolatore richiede una pianificazione un po' maggiore da parte dell'operatore, perché dopo 20 o 25 righe di display, le informazioni cominciano a *uscire dal lato superiore dello schermo* e per l'utente sono perdute.

Terminali Crt più evoluti possono essere programmati

per proiettare una griglia per l'immissione di dati. Ad esempio sullo schermo può essere riprodotto il modulo per un ordine di acquisto. L'operatore, molto semplicemente, *compila* un modulo per ogni ordine d'acquisto, come se si trattasse di un modulo stampato su carta. Poi, dopo che le informazioni sullo schermo televisivo sono state controllate e stampate, l'operatore preme uno o più tasti e le informazioni vengono trasmesse dal Crt al calcolatore. Dopo la trasmissione si cancellano le informazioni sullo schermo lasciando il modulo ordini d'acquisto *di fondo* intatto, per cui si ha la possibilità di far entrare l'ordine successivo. Lo stesso vale per lettura di informazioni da parte del calcolatore e per l'aggiornamento dei dati.

Altri tipi di Crt consentono il display grafico dei dati. Questi tipi di terminali, anziché essere orientati su caratteri e linee, dividono i propri schermi in piccolissime aree denominate punti. I dati da rilevare producono delle curve sullo schermo. Contemporaneamente si possono rappresentare delle informazioni, il cui stadio finale può essere riprodotto su carta mediante una stampante collegata al terminale che fornisce le rappresentazioni grafiche.

Un'altra apparecchiatura d'uscita è la *stampante*. Si tratta di apparecchiature ad alta velocità (da 600 a 2000 righe al minuto) usate per produrre elenchi troppo voluminosi per la stampa sul terminale dell'utente. Nel campo delle stampanti si sono avuti nuovi sviluppi che hanno portato a stampanti di pagine in grado di stampare una pagina intera di informazioni alla volta, con capacità di stampare l'equivalente di 44.000 righe al minuto.

Oggi l'area della stampa delle informazioni in uscita sta concentrando su di sè molta attenzione. Il mondo degli affari e gli organi di governo impiegano calcolatori per scrivere lettere ed altri documenti che, all'apparenza, sembrano stampati manualmente su comuni macchine da scrivere. Il termine globale per questo campo è *elaborazione della parola* (word processing). Uno dei requisiti principali è l'alta qualità della stampa. Regolarmente si assiste alla reclamizzazione di nuovi prodotti che trattano l'elaborazione della parola più velocemente, meglio e a minor costo. L'uscita dell'elaborazione della parola consiste di lettere, fatture e bande pronte la composizione.

Un'altra apparecchiatura ancora è il *plotter*. Il plotter

fornisce, fondamentalmente, la stessa produzione d'uscita del Crt a rappresentazione grafica, ma può produrre grafici molto più grandi e in vari colori. I grafici sono prodotti su carta mediante penne, e penne a sfere, guidate dalla logica di un programma.

L'apparecchiatura a distanza Rje (*Remote job entry*) è un elemento periferico con capacità di interfaccia remota con un calcolatore, nel quale vengono immessi dati. L'operazione viene effettuata mediante la trasmissione su nastro o scheda da un ufficio che può essere ubicato a migliaia di chilometri di distanza dal calcolatore al quale si accede. Reciprocamente, apparecchiature Rje possono emettere i risultati di programmi effettuati mediante trasmissione su nastro o stampante ad alta velocità. Tutte le trasmissioni fra stazioni Rje e calcolatore si effettuano su linee telefoniche. Il termine *linee* non significa necessariamente *cavi*; oggigiorno le unità Rje coinvolgono di norma collegamenti per trasmissioni in microonde o microonde-satellite.

L'interazione hardware/software

Con tutto questo hardware periferico che circonda i calcolatori, ci si domanda: come fanno i calcolatori a sapere ciò che devono fare? Bene, lo scopo del software è questo. Software è la parola che comprende tutti i programmi di un calcolatore. Questi programmi sono scritti da programmatori i quali *comunicano* con un calcolatore mediante una o più apparecchiature d'entrata e uscita, descritte nella sezione precedente. Il programmatore dà istruzioni al calcolatore di usare l'hardware in determinati modi e di seguire la logica fornitagli. Il calcolatore cercherà di fare ciò che gli ha chiesto il programmatore, indipendentemente dal fatto che sia *giusto* o *sbagliato*. Ho ricordato prima che un calcolatore è intelligente quanto il programma che gli viene chiesto di eseguire. Da solo, un calcolatore è un agglomerato di circuiti, fili e altri mezzi elettrici assortiti, incapaci di dirvi persino che 1+1 fa 2.

Come abbiamo visto, un calcolatore è costituito fondamentalmente da tre componenti separate: un'unità centrale di elaborazione, una memoria, un magazzino. Queste tre entità sono sempre presenti, sia che si tratti di un grande calcolatore sia di un minicalcolatore. Alcuni calcolatori molto piccoli non hanno il magazzino permanente: immagazzi-

nano tutto nella memoria. Per questo motivo, ogni volta che si ricorre a un calcolatore di questo tipo, programmi e dati devono essere prima caricati in memoria. Il calcolatore programmabile è un esempio di questo tipo.

Come per le automobili, la differenza fra un modello di calcolatore d'uso generale e un altro, *sottigliezze* a parte, sta nella velocità e nella dimensione. La velocità di un calcolatore viene normalmente misurata in tre aree: tempo necessario all'unità centrale di elaborazione per eseguire un ordine, velocità operativa della memoria (detto anche tempo del ciclo della memoria), tempo impiegato per leggere e scrivere dati da e su un'unità di memorizzazione. Come già esposto, la dimensione si misura come l'entità dello spazio disponibile in una memoria per l'immagazzinamento temporaneo. La dimensione può riferirsi anche all'entità dell'immagazzinamento permanente disponibile sui dischi.

In sintesi i dischi sono unità di immagazzinamento permanente, mentre la memoria è un magazzino temporaneo. I programmi sono *letti* da un disco e *scritti* su una memoria al momento dell'uso, e scritti da una memoria su un disco, sotto controllo di un programma e dell'unità centrale di elaborazione, ogni volta che il programmatore vuole che il suo programma e i suoi dati siano immagazzinati in modo permanente. I dischi immagazzinano i dati magneticamente; la memoria immagazzina i dati elettronicamente. Quando i dati sono scritti su disco, gli impulsi elettrici provenienti della memoria vengono *trasformati* in particelle magnetiche e depositati sul disco della testina di lettura-scrittura. A differenza del braccio di un giradischi, le testine di lettura-scrittura non toccano mai la superficie del disco; esse galleggiano sul disco per una frazione di millimetro.

Perché abbiamo bisogno di immagazzinare sia su nastri sia su dischi? Come già accennato, i *metodi di accesso* utilizzati dal calcolatore per leggere e scrivere dischi e nastri sono diversi. Mentre un calcolatore può leggere e scrivere dati su nastro solo sequenzialmente, a un disco si può accedere casualmente. Ad esempio, un programma immagazzinato su un nastro deve essere letto nella memoria del calcolatore perché il calcolatore possa eseguirlo. Se capita che questo programma è l'ultimo del nastro, il calcolatore deve leggere tutto quanto è registrato sul nastro, prima di trovare il programma richiesto. Deve fare ciò sequenzialmente; non esiste

alcuna strada per *saltare* il nastro. La ricerca di un dato programma può richiedere dieci minuti o più, a seconda della velocità dell'unità. Ciò, in molte installazioni, non è accettabile. Lo stesso programma immagazzinato da qualche parte su un disco, può essere localizzato dal calcolatore e scritto su una memoria nel tempo di millisecondi, perché, anche se originariamente è stato scritto sull'ultima *unità* del disco, il calcolatore sa come trovarlo quasi istantaneamente. Il calcolatore esegue un certo numero di controlli per trovare dove sono immagazzinati i dati sul disco e la testina si porta direttamente sui dati per scrivere le informazioni desiderate sulla memoria.

D'altra parte i nastri sono un mezzo molto conveniente e poco costoso per immagazzinare dati *fuori linea*. Non sempre è necessario disporre immediatamente (*in tempo reale*) della massa dei dati collegati ad un sistema che quindi vengono conservati meglio su nastri.

Come fa un programmatore a sapere dove sono ubicati i suoi dati, siano essi su disco o su nastro? Come fa a sapere dove scrivere i dati su una memoria una volta *trovati*? Come fa a dire a una unità centrale di elaborazione che ha bisogno che si faccia qualcosa? La risposta è che lui queste cose non le sa e quel che più conta, non lo interessano nemmeno.

Ogni calcolatore ha un programma *di fondo (back ground)* sempre attivo. Questo programma di fondo, chiamato con molti nomi, ma al quale di solito si fa riferimento come *programma di controllo, sistema operativo* oppure *sistema esecutivo*, controlla e dirige tutto quanto viene portato avanti nel calcolatore. Assegna spazi per dati sul disco (e si ricorda dove), dice alla memoria di prendere dati dal disco, decide dove c'è spazio nella memoria per questi dati; è il *vigile* universale del sistema. Tutto quanto è in corso all'interno del calcolatore, in qualsiasi momento è guidato dal programma di controllo, che solitamente è il programma più grande e più complicato; è sicuramente il più importante, perché se mancasse, anche il calcolatore più sofisticato sarebbe soltanto un mucchio di fili, transistor e metallo, assolutamente inadeguato a risolvere qualunque problema.

All'inizio, i calcolatori non erano provvisti di programmi di controllo. Si poteva eseguire solo un programma per volta e le risorse del calcolatore (l'unità centrale di elaborazione, la memoria, il disco) non erano mai utilizzate al massimo. Le

attività commerciali e finanziarie dovevano affrontare la prospettiva non allettante di comperare sempre più calcolatori per smaltire il carico di lavoro in crescita continua. L'evoluzione dei programmi di controllo ha cambiato tutto. Oggigiorno i lavori possono essere fatti in concomitanza, se non simultaneamente, e tutte le risorse del calcolatore sono utilizzate più efficacemente. Dati e programmi di molti lavori possono risiedere nella memoria allo stesso tempo e, nonostante che il disco e l'unità centrale di elaborazione facciano una sola cosa per volta, la velocità con cui quell'unica cosa vien fatta fa credere che i lavori siano eseguiti tutti insieme.

Come fa un programmatore a ricordare tutte queste interazioni? Come fa a tenersi al corrente su ciò che deve fare successivamente? Ebbene, in effetti non ha bisogno di sapere. È come schiacciare l'acceleratore: l'unica azione che fate consiste nel premere il pedale dell'acceleratore; il motore fa aumentare l'immissione di benzina nei cilindri, modifica la miscela aria-carburante per mezzo del carburatore, e così via. La stessa *automazione* si ha nei calcolatori. Il programma di controllo farà da *istruttore* del programmatore. A volte le cose possono complicarsi: per esempio, non sempre i dati su un disco sono consecutivi, le informazioni possono essere divise in sezioni su varie parti del disco. Un compito del programma di controllo è quello di trovare i dati per il programmatore e fargli credere che tutte le informazioni sono quelle di un grande *file* di dati. L'unica cosa che il programmatore deve fare è dare istruzioni al calcolatore, in linguaggio di programmazione, di leggere i dati. Il programma di controllo farà il resto.

Ho appena usato la parola *file*. Che cos'è un *file*? Semplice, un *file* si compone di dati logicamente correlati e considerati come componenti di un insieme di informazioni. L'insieme dei nomi e degli indirizzi degli abbonati a un periodico può costituire un *file* di dati. Un *file* è sempre ripartito in registrazioni. Un *file* può contenere una registrazione o diversi milioni di registrazioni. Una registrazione consiste in dati associati a ciascuna unità separata nel *file*. Le informazioni relative a nome e indirizzo della sig.na Bianchi, insieme alle altre informazioni attinenti al suo abbonamento, si considerano come una unità, o registrazione. Le stesse informazioni riguardanti il sig. Brambilla costituiscono un'altra registrazione. Il nome della sig.na Bianchi costituisce un *campo* al-

l'interno della registrazione; la via in cui abita costituisce un altro campo della stessa registrazione, anche la città, lo stato e il c.a.p. sono, ciascuno, campi separati della registrazione.

Quando un *file* comprende informazioni di questo tipo (indirizzi, informazioni su parti di un sistema di inventario, risposte a questionari di una rilevazione) le registrazioni collettive sono chiamate *file* di dati o base di dati. Quando un programmatore scrive il codice di un programma, scrive anche un *file*. Infatti, il mezzo di immagazzinamento del calcolatore non è in grado di dire la differenza fra un tipo e un altro di *file*; è responsabilità del programmatore sapere quali *file* sono stati creati e a quale scopo.

All'interno dello stesso *file* i dati possono essere fisicamente divisi in varie parti, nel momento in cui sono immagazzinati su disco, perché il programma di controllo cerca di utilizzare quanto più possibile gli spazi sul disco. Diciamo che un *file* è immagazzinato in qualche parte di un magazzino permanente. Successivamente si decide di aggiungere alcuni dati al *file*. Per fare ciò, si chiede al calcolatore, mediante un programma, di *aprire il file*. Ora i dati entrano e vengono scritti sul disco, in aggiunta al *file* già esistente. Adesso però, si tratta di immagazzinare più informazioni di quelle precedentemente contenute nel *file*. Il calcolatore dove mette i dati aggiuntivi? Per prima cosa il programma di controllo deciderà se i dati si possono ancora inserire nella sezione originale del disco; in caso affermativo si aggiungeranno i nuovi dati. In caso contrario, si deve trovare spazio da qualche altra parte; il programma di controllo dirà al calcolatore di rompere i dati in parti, in modo da inserire ogni cosa nel disco. Per realizzare questa condizione, il programma di controllo potrebbe ritenere necessario usare spazio in aree diverse del disco, per cui il *file* verrà immagazzinato in pezzi, ovvero *frammentato*. Ma ciò non è un motivo di allarme: il calcolatore ricorderà ciò che ha fatto.

Il programmatore non ha niente a che vedere con tutte queste operazioni: *preme il piede sull'acceleratore*. Tuttavia, un buon programmatore, prima di immagazzinare una base dati, dovrà interessarsi ai problemi di spazio, perché l'accesso ad una base dati frammentata richiederà al calcolatore (che dovrà impegnarsi maggiormente nella ricerca) più tempo e più risorse che non l'accesso a una base dati *contigua*. Sfortunamente, nei casi in cui le basi dati cambiano spesso,

non è sempre possibile controllare questa situazione e immagazzinare dati efficacemente. Ad esempio, una base dati di 100 nomi oggi, potrebbe averne 200 domani. Il programma di controllo cercherà di immagazzinare questi dati aggiuntivi *vicino* alle informazioni già esistenti nel disco, ma se qualche altro *file* occupa già quello spazio, non ci sarà altra scelta se non quella di cercare spazio da qualche altra parte.

Alcuni sistemi trattano problemi come questo non proprio così automaticamente e il programmatore deve preoccuparsi di pre-assegnare lo spazio che il suo *file* occuperà. Tuttavia l'operazione, anche se si rende necessaria, è abbastanza semplice.

Le prestazioni generali del calcolatore sono favorite se, a intervalli regolari, il dipartimento operazioni obbliga i dati su disco a diventare contigui. Questo procedimento viene chiamato *ricarica del disco* e implica la trascrizione di tutti i dati dal disco su un nastro (dove i dati diventano contigui), quindi la *pulizia* dei dischi e la nuova copiatura dei dati dai nastri che erano stati creati. Questa operazione è spesso evitata, perché richiede molto tempo e interrompe il flusso del lavoro di routine. Il procedimento, peraltro, è molto utile dato che migliora il rendimento del calcolatore (cioè l'efficienza e la velocità con la quale un calcolatore tratta l'entrata, i calcoli e l'uscita). Si recupera, inoltre, e i riutilizza sui dischi lo spazio non usato dal calcolatore nel corso delle trattazioni di *file* giorno per giorno.

In questo momento potreste essere un po' confusi. Quali dati risiedono sui dischi e quali nella memoria?

Abbiamo visto che i dischi sono mezzi di immagazzinamento permanente. Tutti i *file* e i programmi sono conservati lì. Quando si deve accedere a un *file*, questo viene letto dal disco ed inserito nella memoria. Se il *file* si identifica con un programma, potrebbe aver bisogno, quando *funziona* o *esegue*, di informazioni da una o più basi dati. Questi dati saranno letti anche nella memoria. Se la base dati è piccola, la memoria può leggerla immediatamente. Se la base dati è grande, la memoria leggerà un segmento solo dei dati, forse una registrazione appena, la tratterà e, finita l'operazione su questa prima registrazione, leggerà quella dopo e così via.

La capacità di immagazzinamento di una memoria di solito è molto inferiore a quella di un disco e allo stesso tempo più costosa. Il programmatore deve sapere di quali dati il

suo programma ha bisogno e darà istruzioni al programma perché prenda quei dati dal disco e li metta nella memoria. Il programmatore, ovviamente, deve operare entro i confini della disponibilità dello spazio di immgazzinamento della memoria. Deve essere sicuro che non si accinge ad occupare nella memoria più spazio di quello che ha a disposizione. Se cerca di utilizzare più spazio di quello che c'è, il calcolatore glielo dirà e il programma non verrà svolto.

La memoria ha un limite finito di disponibilità di spazio, però, se vi è una certa scarsità di memoria, l'acquisto di più memorie non è necessariamente l'unica risorsa disponibile. Spesso esistono diversi metodi seguendo i quali un programmatore può ottenere i risultati desiderati da un dato programma. *Destreggiandosi* con i dati fra disco e memoria (il che significa che parti più piccole dei dati risiedono nella memoria, una alla volta) in certi casi si può superare questa limitazione di spazio. D'altra parte, forzare di più l'interazione fra dischi e memoria ritarda un po' il processo di calcolo, perché l'unità centrale di elaborazione dovrà lavorare di più e, a un certo punto, un programmatore potrebbe trovare che nel risolvere il problema della carenza di memoria ha strozzato l'unità centrale di elaborazione. Se dovesse verificarsi una cosa del genere, è meglio riprogrammare in modo da usare di più la memoria e meno l'unità centrale di elaborazione, perché l'acquisto di più memorie è forse meno costoso che comperare una nuova unità centrale di elaborazione.

Vediamo un esempio di interazione disco/memoria. Una società editrice dispone di una base dati nel suo magazzino permanente del calcolatore con informazioni sugli abbonati a un periodico di sua pubblicazione. Ciascuna registrazione della base dati contiene nome e indirizzo di un abbonato. È necessario disporre di un programma in grado di modificare nomi e indirizzi conservati in questo *file*. L'operatore scrive un programma che legge la registrazione da modificare nella base dati e la mette in memoria. Ora il programma chiede all'operatore informazioni sul nuovo indirizzo. Queste nuove informazioni *sostituiscono* la parte di indirizzi della registrazione che è in memoria. Dopo aver fatto ciò, il programma dà istruzioni al calcolatore perché scriva di nuovo la registrazione sulla base dati del disco. Situazione semplice, ma anche in presenza di un problema semplice (e tutte le cose che si fanno con i calcolatori sono piuttosto semplici) i programma-

tori possono (e ciò accade) commettere degli errori. Il software è il *guidatore*: fa svolgere al calcolatore il compito che ci si aspetta da lui. L'analogia con le automobili è vera ancora: la macchina da corsa più potente, è valida quanto il guidatore che la porta; da sola, senza guidatore, sembra soltanto bella.

Le alternative del manager

Supponiamo che siate il presidente di una società che dispone di un calcolagore tipo A. Il manager Edp un giorno viene a trovarvi e vi dice che si rende necessario *potenziare* il calcolatore. Cosa fate? Ovviamente, potreste decidere di non far proprio niente, ma il vostro manager Edp brontolerebbe e di sicuro prima o poi (forse prima che poi) trovereste: che le relazioni che vi occorrono non vi vengono consegnate in tempo, che il dipartimento Edp ha registrato un mucchio di ore di straordinario e, in generale, che il lavoro d'ufficio non sta andando molto bene. Evidentemente dovete fare qualcosa per raddrizzare la situazione. Ma il potenziamento del calcolatore è costoso, per cui prima di impegnarvi nella spesa avete bisogno di risposte ad alcuni quesiti. A proposito di costi, si tratta di cifre dell'ordine delle decine e centinaia di migliaia di dollari, qualche volta anche di milioni. Quali domande farete per rendervi conto di quanto è necessario?

Per prima cosa non dovete pensare che il calcolatore, al momento, svolga tutto il lavoro in condizioni ottimali. È noto, infatti, che nella gran parte delle installazioni un quarto, mediamente, di tutta la potenza di un calcolatore non viene pienamente utilizzata. L'espressione *non viene pienamente utilizzata* sta ad indicare che il vostro calcolatore potrebbe fare un lavoro più produttivo di quello che fa attualmente, se si seguissero certe direttive.

Se dovete risolvere un problema di carenza di immagazzinamento permanente e vi si chiede di impegnare la vostra ditta per una somma di 30-40.000 dollari per una nuova unità dischi, chiedete al manager Edp di quantificare l'aliquota di magazzino che, al momento, non necessariamente deve essere *in linea* di continuo. Vecchie, vecchissime edizioni di programmi, le basi dati del mese scorso possono essere prese molto agevolmente dal magazzino permanente e messe su nastro magnetico per la conservazione. E che dire dei risultati delle analisi vendite, elaborati la settimana scorsa? L'unico

motivo per cui rimangono sul disco è perché la stampante, quando è stato fatto quel programma, era occupata e quindi i dati in uscita sono stati dirottati sul disco per essere stampati nel momento più idoneo. Anche quel programma può essere cancellato. La gente tende a trascurare un po' il magazzino; è lì disponibile e così ci si abitua. Non vi è nulla di negativo in ciò ma quando giunge il momento in cui vi si chiede di spendere denaro per altre unità disco, prima di tutto si devono esaminare gli elementi ai quali ho accennato.

Può anche darsi che vi si dica che siete *a corto di memoria*. Bene, ma perché il mese scorso non eravate a corto di memoria? Forse è stato scritto un nuovo programma per affrontare il problema XYZ ed ora non c'è posto abbastanza nella memoria per contenere questo nuovo sistema superingegnoso. Voi volete che venga fatto *girare* giornalmente, vero? Prima di fare qualunque cosa, scoprite se questo nuovo sistema è scritto (programmato) in modo ottimale. Ha effettivamente bisogno di tanta memoria quanta ne usa ora? I programmi possono essere scritti in modi molto diversi e fornire ugualmente gli stessi risultati. Un'altra analogia: se voglio andare in macchina da Milano a Bologna posso prendere l'autostrada del Sole, un itinerario abbastanza diretto. Potrei prendere però la statale che passa per Piacenza ma è un giro vizioso. Il paesaggio è bello, però impiego più tempo. La stessa cosa vale per i programmi che vengono gestiti da un calcolatore. Un programma può prendere molte piste diverse, senza modificare i risultati. Alcune piste seguono un tracciato più diretto di altre.

Considerate questo esempio. Supponiamo che abbiate bisogno di un programma che vi dia il risultato dell'espressione 1+1=..... Un approccio al problema è il seguente: immagazzinate il valore 1 nella parola della memoria. Dopo il calcolo, immagazzinate ancora il risultato in un'altra parola della memoria. Il segno = è usato non come un segno "uguale", ma come segno di "sostituzione". Poniamo I = 1 significa: mettete il valore 1 nell'ubicazione della memoria I, sostituendo qualunque contenuto che l'ubicazione I della memoria aveva prima di questa operazione. Il programma avrebbe questa configurazione:

```
PONIAMO I = 1
PONIAMO J = 1
PONIAMO K = I + J
```

Il risultato immagazzinato nella parola della memoria che ho chiamato K, è senz'altro corretto. Però per risolvere il problema, ho utilizzato tre parole della memoria. Ora riscrivo il programma:

PONIAMO I = 1
PONIAMO I = I + I

Ciò significa: mettete il valore 1 nella parola della memoria I, poi cambiate il valore di questa parola che abbiamo chiamato I in modo tale che contenga il risultato dell'addizione del presente valore della parola al presente valore della parola. Il risultato di questo secondo programma sarà lo stesso di quello prodotto dal primo programma, ma il secondo programma utilizza solo un terzo delle parole della memoria.

Ma come fate a saperlo? A chi chiedete di scoprire un fatto del genere? Le risposte possono essere apprese da chi ha disegnato il sistema. Lui è il dente più importante nella ruota della programmazione. Fate che dia un'altra occhiata a questo nuovo sistema, e se è preparato, si guadagnerà il suo peso in oro e più.

Avete bisogno di un'unità centrale di elaborazione più veloce? Può darsi, forse non ci sono abbastanza ore nella giornata per eseguire tutti i programmi e le relazioni, per cui un'unità più veloce, che fa i calcoli più in fretta può porre rimedio alla situazione. Ma guardate alle ore di uso effettivo del vostro calcolatore. Se funziona dalle 8.30 alle 17.30, perdete 15 ore preziose ogni giorno. Il calcolatore è acceso comunque (molti costruttori consigliano di non spegnerlo, perché i picchi dell'energia elettrica che si creano per *tirarlo di nuovo su* possono danneggiare la delicata elettronica), e pertanto perché non sfruttare quelle ore? Disporre di uno staff operativo che lavori di notte, probabilmente è meno costoso che comperare una nuova unità centrale per l'elaborazione. Oppure si può chiedere ad alcuni programmatori di lavorare di sera, e scrivere e provare i programmi.

È tipico che gran parte del tempo di un calcolatore sia impiegato per modificare e accrescere programmi esistenti o scriverne di nuovi. Un impiego del genere è esattamente *tempo di sviluppo* e non porta a risultati. La sezione operativa fa uso del *batch differito*? Il batch differito è un mezzo mediante il quale i lavori sono preparati per essere eseguiti

durante il giorno. I lavori non vengono effettivamente eseguiti fino a dopo una certa ora, in modo da distribuire su tutte le 24 ore le risorse disponibili del calcolatore, anziché nelle otto o nove ore del turno giornaliero. Il segreto sta nell'utilizzazione totale delle apparecchiature disponibili.

È necessario disporre di una nuova unità nastro solo perché si dà il caso che il nuovo super-programma abbia bisogno di un'unità in più? Forse ciò è vero, ma prima di decidere l'acquisto dell'unità nastro, valutate se si può rendere disponibile dello spazio sul disco per far quel tipo di programma, ovviando all'inconveniente di un altro nastro. Normalmente ciò richiede solo una semplice modifica del programma. In realtà, potreste voler fare un altro passo indietro e chiedere se il nuovo programma è proprio necessario. Potrebbe non essere così.

Come vedete, l'utilizzazione ottimale di un calcolatore è senza alcun dubbio un problema di programmazione. Spesso per una ditta è un eccellente investimento avvalersi di un consulente non coinvolto nei problemi quotidiani dell'organizzazione, che perciò può vedere il quadro generale, non solo il particolare, per studiare le esigenze della ditta a livello di calcolatore e armonizzarle con l'hardware disponibile. Ora però avete una conoscenza sufficiente per cui, se siete nella condizione di approvare o disapprovare delle spese per l'hardware, non sarà il caso che vi facciate intimorire da poche persone *addette ai lavori*. Un consiglio: se non capite di che cosa parlano quelle persone (ed è facile, perché spesso sembra preferiscano che voi non capiate) ditelo chiaramente e fatevi spiegare il problema in un linguaggio comprensibile. Non vi è assolutamente niente di *mistico* nei calcolatori. Di nuovo è come con le macchine: conoscete le funzioni del motore, del cambio, del tergicristallo, e così via, ma è necessario che conosciate i dettagli intricati di ciascuna di queste componenti per guidare una macchina? Naturalmente no.

Ora siete orientati sulle operazioni interne del calcolatore commerciale. Tutti i costruttori di calcolatori cercano di realizzare la miglior macchina, questo ovviamente crea un mercato vantaggioso, competitivo per il compratore di hardware. Oggi vi sono ampie possibilità di scelta di apparecchiature in grado di soddisfare le esigenze di pressoché tutti gli operatori. Però il compratore deve essere attento. Alcune imprese comprano esclusivamente *marche di nome*,

solo perché conoscono il nome del costruttore; altre sono più avventurose e comprano apparecchiature meno note. Da qualunque costruttore una ditta decida di comperare, ci si deve assicurare attentamente che il calcolatore soddisfi l'esigenza, che non sia *troppo* grande, nè la macchina sbagliata per lo scopo al quale è destinata. Purtroppo sono poche le ditte che intraprendono uno studio completo delle proprie esigenze prima di decidere l'acquisto di un calcolatore.

Apparecchiature e servizi

Per darvi un'idea di ciò che oggi è disponibile, consideriamo i vari tipi di apparati e servizi che fanno a gara per richiamare l'attenzione degli acquirenti.

Vi sono minicalcolatori, calcolatori di media e grande potenza, calcolatori tipo *time-sharing*, calcolatori con trattamento distribuito delle informazioni e infine ditte specializzate che soddisfano le esigenze di elaborazione della clientela, ricorrendo all'elaborazione differita dei programmi. Descriverò il significato di questi termini.

Il significato di minicalcolatore è implicito nel termine: si tratta di piccoli calcolatori. Sono più piccoli e dotati di minore potenza dei grandi calcolatori, nonostante alcuni mini abbiano così tanti *hang-ons* (apparati aggiunti) da raggiungere quasi la potenza dei grandi calcolatori. Spesso i mini rappresentano un ottimo investimento per le piccole imprese a cui serve una gamma limitata di programmi o con volumi di dati da elaborare relativamente piccoli. Spesso i mini possono essere gestiti, azionati e programmati da una persona soltanto, con conseguenti favorevoli riflessi sui costi del personale.

Alcuni mini eseguono solo un insieme predeterminato di programmi. Alcuni costruttori producono minicalcolatori che trattano una gamma completa di programmi contabili, inclusi i saldi debitori e creditori, retribuzioni, partitario generale per imprese piccole-medie. Se la vostra ditta ha problemi comuni a molte altre, è facile trovare dei minicalcolatori *pronti all'uso*, con programmi già scritti e installati.

Una certa cautela s'impone: le necessità degli utenti sono differenti e se un sistema pronto all'uso non vi soddisfa appieno potreste accorgervi che riscrivere programmi per avvicinarvi alle vostre esigenze comporta un notevole investi-

mento in tempo e costi. Spesso, in questi casi, è meglio partire da zero. Generalmente i minicalcolatori hanno prezzi decisamente inferiori rispetto a quelli di acquisto o di affitto dei grandi. Se poi potete utilizzare i programmi già pronti non avrete neppure bisogno dei programmatori con evidenti risparmi di budget. Il funzionamento dei minicalcolatori con programmi già pronti è di facile apprendimento da parte di chi lavora in contabilità e si può quindi evitare la formazione di strutture apposite. I mini, inoltre, funzionano generalmente molto bene a temperatura ambiente; l'impianto per l'aria condizionata di cui già disponete è sufficiente a tenere la macchina a temperatura normale di lavoro, contrariamente ai calcolatori più grandi che richiedono sistemi di aria condizionata potenti, installati *ad hoc* per il raffreddamento.

Il mini è la macchina per la vostra ditta se è sufficientemente grande da soddisfare le vostre esigenze e, nel caso dei modelli preprogrammati, se potete reprimere un po' i vostri desideri in modo da adeguarvi alle regole dei programmi disponibili. Molte ditte fanno così e per loro i mini sono un ottimo investimento. Oltre ad essere impiegati da società di piccola-media grandezza, spesso i minicalcolatori vengono acquistati da ditte di progettazione, da consulenti fiscali e altre società che hanno bisogno di eseguire notevoli quantità di calcoli. Molti mini calcolatori del giorno d'oggi sono più potenti dei grandi calcolatori degli anni 1950 e 1960.

Altre società, troppo grandi per il minicalcolatore o con esigenze di molte applicazioni diversificate si orienteranno, per la soluzione dei loro problemi, verso calcolatori più grandi. Ma i costi iniziali sono estremamente alti, specie se una ditta non dispone già di uno staff per la programmazione, per cui spesso si prende in considerazione una soluzione intermedia che, come qualche volta accade, diventa permanente. Questa soluzione viene denominata *time-sharing*.

In questo caso non si andrà all'acquisto del calcolatore, ma la società che intende utilizzare il calcolatore in *time-sharing* dividerà *il tempo di utilizzazione* di un grande calcolatore di una ditta che vende servizi *time-sharing*. I calcolatori di queste società (ne esistono molte che si avvalgono di reti nazionali e internazionali) sono macchine grandi, molto potenti, e svolgono programmi che vanno dal singolo utente a 200 utenti diversi nello stesso tempo. In effetti, i programmi di questa massa di utenti non vengono svolti esattamente

nello stesso tempo, ma l'utente non se ne rende conto. I programmi di controllo che seguono il funzionamento dei calcolatori in *time-sharing* sono estremamente sofisticati, in grado di tenersi al corrente di tutto ciò che accade agli utenti collegati, cosicché ciascuno pensa di essere l'unico a usare il calcolatore, mentre in realtà può trovarsi ad utilizzare il sistema unitamente a molti altri utenti, ognuno dei quali fa qualcosa di diverso.

Un termine migliore per il *time-sharing* è quello di elaborazione dati in linea. Il vantaggio principale di questo metodo operativo consiste nel fatto che il vostro lavoro viene immediatamente elaborato in linea: di scarsa o nessuna rilevanza, il fatto che condividete un calcolatore con altri utenti. Un altro grande vantaggio del calcolatore in *time-sharing* in linea sta nel fatto che l'utente può collegarsi in qualsiasi momento lo desideri componendo il numero di telefono del calcolatore addirittura dal proprio ufficio e, continuando poi a creare od eseguire il proprio programma su un *terminal* con tastiera o video.

L'utilizzazione di molti calcolatori installati nelle aziende è controllata da un apposito settore che solitamente assegna ai programmi dell'utente una *aliquota* del suo programma giornaliero. L'utente, ovunque si trovi, dovrà aspettare da un'ora a un paio di giorni, prima che i risultati del suo programma siano noti. Impiegando un calcolatore in *time-sharing*, fra macchina e utente si instaura invece un'interazione immediata e costante. I terminali *time-sharing* operano a velocità che raggiungono i 120 caratteri per secondo. Alcune ditte di *time-sharing* offrono servizi con velocità addirittura di 480 e 960 caratteri per secondo, che equivale a 11.500 parole al minuto dattiloscritte a mano.

Per quanto riguarda il costo, se si sommano le spese connesse ad un calcolatore in proprio, quali l'hardware in sè, la manutenzione, lo spazio occupato, il sistema speciale per il condizionamento dell'aria e, non ultimi affatto i costi del personale addetto alla programmazione e al funzionamento delle macchine (i costi del personale, di regola, ammontano al quadruplo di quelli dell'hardware), il *time-sharing* può senz'altro ritenersi per molte imprese un metodo economico. Specialmente ora che i venditori di *time-sharing* sono in grado di fare sconti sul listino prezzi degli *utenti casuali* a ditte che intendono usare risorse considerevoli per molto tempo.

Altre possibilità che molte grandi ditte di servizi *time-sharing* offrono ai clienti sono rappresentate dalle convenzioni per la gestione di apparecchiature, mediante le quali una società acquista le parti di uno o più calcolatori. Questi possono essere dislocati sia negli uffici del cliente, sia presso la ditta di servizi *time-sharing*, che curerà la manutenzione dell'hardware e del software del sistema. In effetti per molte società che attualmente dispongono di installazioni proprie, è giunto il momento di prendere in considerazione il passaggio alla locazione di questi servizi da una ditta di *time-sharing*. Molte imprese, impegnate nella fabbricazione e vendita di prodotti, hanno già abbastanza problemi sul mantenimento o ampliamento della loro fascia di mercato, da non doversi preoccupare ulteriormente della gestione di altre *linee di prodotto* della sezione Edp. Oltre a ciò non va dimenticato che il passaggio ad accordi di *time-sharing* comporta spesso un risparmio di denaro. Finora il *prestigio* di avere un calcolatore in proprio ha avuto la meglio sulle realtà di una gestione valida. In generale sarebbe opportuno che le imprese riprendessero in esame le decisioni prese riguardo ai calcolatori, alla luce delle odierne possibilità di evoluzione e di espansione.

Molti venditori di *time-sharing* forniscono servizi di programmazione e con cifre accettabili viene programmata la soluzione dei problemi dell'utente. Dal momento in cui viene deciso un accordo di questo genere, la ditta di *time-sharing* si impegna a consegnare un prodotto finito pronto da gestire, così che, per conseguire i risultati desiderati, la società utente non ha alcun bisogno di personale specializzato in calcolatori e programmazione. Anche se i corrispettivi sembrano piuttosto salati, confrontati con i costi interni d'impiego di analisti, esperti programmatori, la cosa si rivela conveniente.

In genere i gruppi di lavoro di ditte affermate di servizi di *time-sharing* operano molto bene e realizzano sistemi molto più in fretta di quanto ci si possa aspettare dal proprio servizio Edp, spesso sotto pressione per via del tempo. Il lavoro delle ditte di servizi di *time-sharing* è sotto garanzia, per cui gli errori involontari di logica o programmazione vengono subito corretti a norma di contratto.

Nel caso di rapporti con una ditta di servizi *time-sharing* occorre tener presente due aspetti:

1) prima di firmare un contratto dovete conoscere gli

importi esatti del costo di sviluppo del sistema, e dell'impiego mensile del calcolatore;

2) prima di scrivere le istruzioni relative al sistema, sottoscrivete (e procurate di averla prima della società di *time-sharing*) *una specifica sul lavoro che deve essere eseguito.* Nella dichiarazione devono apparire in dettaglio tutte le parti del sistema, le dimensioni e le configurazioni delle registrazioni di tutti i *file* di dati e dei report, l'interazione logica fra tutte le componenti del sistema. Per evitare contrasti fra utente e società di servizi *time-sharing*, nella dichiarazione dovranno essere descritti esattamente i servizi che devono essere forniti.

L'attenzione delle ditte *time-sharing*, verso il mercato si sposta rapidamente. Dieci anni fa, infatti, vendevano la maggior parte dei loro servizi a società di progettazione ingegneristica, che impiegavano i calcolatori per fare calcoli. Oggi queste società utilizzano, per i calcoli, minicalcolatori in proprio, per cui le ditte di *time-sharing* per vendere i loro servizi si sono cercate nuovi mercati. Per riuscire nell'impresa, si dovevano sviluppare nuovi prodotti. Il nuovo settore di intervento è il management che era ed è ansioso di essere più informato: un servizio di riporto dei dati e delle informazioni tempestivo ed esatto è diventato essenziale alla redditività. Di conseguenza le ditte di servizi di *time-sharing* hanno sviluppato sistemi di *data base* per la gestione, il settore finanziario e la programmazione, ed hanno così allargato i propri orizzonti. L'attenzione si è totalmente spostata dalle soluzioni di problemi di calcolo al soddisfacimento di requisiti ben definiti in termini di riporto dati, informazioni e previsioni.

Questa tendenza continua. Le ditte di servizi *time-sharing* hanno sempre maggiori richieste di studi completi sulla gestione, intesi a definire tutta la gamma delle esigenze societarie nell'area dell'utilizzazione dei calcolatori e quindi perfezionare le soluzioni proposte. Oggi molte ditte di servizi *time-sharing* impiegano personale con precedenti commerciali e con esperienza generale, piuttosto che *programmatori nella torre d'avorio* che non hanno pratica del mondo degli affari.

Un altro metodo ancora per l'utilizzazione dei calcolatori è quello delle ditte specializzate nell'elaborazione differita dei programmi. Queste società di servizi sono proprietarie di

uno o più grandi calcolatori e si specializzano in programmi che curano la soluzione dei problemi contabili delle società clienti. Inoltre, hanno profonde cognizioni dell'area della preparazione dati; offrono servizi di perforazione, lettura ottica e verifica dati. A differenza delle ditte di servizi *time-sharing*, dove il cliente può *chiamare* un calcolatore da qualsiasi luogo del paese, o anche del mondo, attraverso reti di comunicazione telefonica complesse e sicure, molte agenzie di servizi adottano il criterio di farsi consegnare i dati d'ingresso e far ritirare gli elaborati d'uscita. La specializzazione delle agenzie di servizi ha consentito loro di essere altamente valide in termini di costi/prestazioni. Nelle aree in cui l'immediatezza del riscontro non è di grande importanza per una società – ad esempio le aree della contabilità generale e delle retribuzioni – le agenzie di servizi soddisfano precise esigenze. Non è una sorpresa che perfino alcune ditte di servizi *time-sharing* si rivolgono ad agenzie di servizi per l'elaborazione dei propri libri paga.

Oltre ai tipi di calcolatori e servizi ricordati, si sviluppano continuamente nuovi prodotti e applicazioni. Tre risultati di questi sviluppi sono: l'informatica distribuita, il terminale intelligente e il microcalcolatore.

L'informatica distribuita consente a una società di distribuire le sue elaborazioni fra un certo numero di calcolatori. Ad esempio, una ditta con molte fabbriche sparse geograficamente può realizzare una rete di elaborazione distribuita, mediante la quale risulta possibile elaborare localmente i programmi della consociata interessata. I dati creati da questa fabbrica, significativi anche per altre fabbriche della catena e per la direzione generale, vengono trasmessi su linee telefoniche al calcolatore principale che così dispone di informazioni relative alla produzione, alle vendite, al magazzino e al personale, a relazioni gestionali e può distribuire queste informazioni ad altre utenze che ne hanno bisogno.

Teoricamente tutto ciò sembra estremamente valido; una fabbrica non deve aspettare che i programmi siano sviluppati dalla direzione generale; non deve attendere i dati giornalieri e in generale, i gruppi locali possono usufruire di ampia autonomia per sviluppare ciò che considerano migliore per le loro esigenze. Nella pratica, però, si corre il rischio che programmi del genere siano sviluppati più volte dalle fabbriche separate, che sorgano problemi circa il tipo e la forma dei

dati da trasmettere al calcolatore della direzione generale e che ciascun manager Edp locale consideri il proprio calcolatore come *un suo regno* e non sia molto soddisfatto nel dover dividere programmi e informazioni con le altre fabbriche. Per ottenere che l'elaborazione distribuita funzioni bene, è necessario che l'ufficio principale metta in atto un rigido controllo gestionale centrale sulle unità separate che costituiscono la catena. Dal punto di vista tecnico, l'avvento dell'elaborazione dati distribuita sembra soddisfare un'esigenza sentita, ma il giudizio, dal punto di vista della gestione, sull'ipotesi di buona riuscita del sistema è ancora in atto.

I terminali *intelligenti* sono mezzi di ingresso e uscita con maggiori possibilità dei terminali di solito impiegati per accedere a calcolatori *time-sharing*. Sono cioè in grado di operare come calcolatori autonomi di limitate capacità. Oltre ad avere una unità centrale e capacità di memoria, i terminali intelligenti hanno capacità di magazzino, permanente, disponibile sia su disco, sia su unità cassetta. I dati si possono introdurre nel magazzino del terminale intelligente usando programmi di entrata che svolgono funzioni di verifica dell'accuratezza dei dati, quali il controllo che i dati entrino nel tracciato grafico giusto ed abbiano grandezza corretta. Dopo che i dati sono stati dattiloscritti, controllati e immagazzinati nel terminale intelligente, questo viene collegato, a mezzo linea telefonica, a un grande calcolatore a cui trasmette i dati stessi.

Uno dei vantaggi di questo approccio consiste nella riduzione del tempo di collegamento con un grande calcolatore, perché la trasmissione dati con mezzi elettronici risulta migliaia di volte più veloce che non la battitura manuale dei dati nel calcolatore. Impiegando meno tempo si risparmia denaro. Un altro vantaggio sta nel fatto che i programmi propri del terminale intelligente sono già stati sottoposti a controlli di validità dei dati da trasmettere al grande calcolatore che li accetta quindi senza successivi controlli. Questa operazione consente, inoltre, al calcolatore maggiore di svolgere altre funzioni per più utenti.

È anche possibile realizzare una rete di elaborazione distribuita, con l'impiego di terminali intelligenti, nonostante in molti casi le capacità di elaborazione del terminale intelligente possano non essere sufficienti a svolgere le funzioni locali richieste al calcolatore di un sistema organico di in-

formatica distribuita. Peraltro i confini fra minicalcolatori – che oggi sono le *colonne* di una rete di elaborazione distribuita – e terminali intelligenti stanno diventando indistinti, come stanno diventando meno definiti i contorni fra minicalcolatori e calcolatori tradizionali.

La seconda attività che i terminali intelligenti sono in grado di compiere bene è quella di produrre documenti autonomamente, usando la propria capacità di immagazzinamento accoppiata alla capacità di programmazione e ad una stampante di alta qualità. I documenti che possono essere generati sono fatture, dichiarazioni e lettere. Ad esempio, le lettere possono essere prodotte quando il terminale intelligente combina un *file* di nomi e indirizzi con altro *file* contenente il testo di una lettera o i testi di lettere diverse e ne risulta un documento formalmente professionale, personalizzato. Inoltre, i dati relativi a nome e indirizzo possono essere trasmessi bidirezionalmente fra terminale intelligente e calcolatore, così da sviluppare elenchi di indirizzi selettivi per spedizioni postali finalizzate a gruppi di specifico interesse.

Tema del momento è la spersonalizzazione o meno di lettere realizzate mediante calcolatore. In realtà, però, le lettere prodotte dal calcolatore non hanno frasi o paragrafi messi a casaccio, in confusione: contengono, invece, espressioni selezionate su vari argomenti effettivamente indirizzate a chi è stata inviata la domanda. I documenti così prodotti possono diventare forti e personalizzati strumenti di comunicazione fra società e clienti, consentendo anche un interscambio maggiore su nuovi prodotti, problemi ed altre aree di reciproco interesse. Grazie a questo sistema, inoltre, si riesce ad ottenere una risposta. Nel passato, istituzioni commerciali e governative erano nella impossibilità di rispondervi presto, ammesso e non concesso che vi rispondessero, perché non avevano il tempo necessario per trovare la persona responsabile, formulare la risposta e dattiloscrivere la lettera. Inoltre le lettere spesso venivano inviate ad un settore sbagliato, ora invece, possono essere inoltrate all'unità competente e il vostro problema ha maggiori possibilità di essere risolto. L'unico motivo d'insoddisfazione di molta gente è che in questa era dei calcolatori, si stanno perdendo i contatti reciproci e le comunicazioni fra le persone ristagnano anziché migliorare. Spesso ciò è vero, specie nel caso di

ditte che hanno sostituito completamente i propri gruppi di interazione umana con i calcolatori. Credo fermamente che questa tendenza possa essere invertita e ciò accadrà non appena le società si renderanno conto che il calcolatore è soltanto uno strumento, non il prodotto finale.

Un'altra area che ha registrato elevati progressi è quella dei microcalcolatori. Il microcalcolatore è la versione miniaturizzata del minicalcolatore. Alcuni modelli ora disponibili costano poco più di un televisore a colori e trovano impiego in varie applicazioni *in casa*. Estratti conto, denunce dei redditi, archiviazione di nomi, indirizzi e numeri di telefono, giochi con il televisore, conservazione di ricette utili: si tratta di aree in cui i microelaboratori sono destinati ad avere un ruolo sempre maggiore. Presto saranno disponibili i programmi che vi insegneranno come programmare il vostro micro, così che virtualmente non vi saranno limiti alle applicazioni della vostra macchina. Altri programmi vi insegneranno qualsiasi argomento immaginabile, con test progressivi forniti dalla macchina stessa. Inoltre presto sarete in grado di collegare il vostro microcalcolatore ad uno grande – ad esempio per avere consigli d'emergenza di tipo medico e legale – oppure di avere un microcalcolatore che esegue le stesse funzioni di un terminale intelligente.

Le possibilità sono infinite e realistiche. Col vostro micro potrete pagare fatture, prenotare gli spettacoli preferiti, sapere la situazione del vostro conto corrente su qualsiasi argomento. Il futuro di questi sviluppi fa trasalire. Presto vi accorgerete che il vostro micro è indispensabile quanto l'automobile.

Adesso avete una buona idea di ciò che è un calcolatore e come funziona. Vi siete resi conto che i calcolatori da soli non sono intelligenti. L'entità che fa sembrare intelligente un calcolatore è il software che vi opera. Il software è creato da analisti, progettisti e programmatori. La gente del mondo dei calcolatori usa una frase che dice: "Garbage in, garbage out" (si abbrevia *Gigo*) che significa: "Se nel calcolatore si inserisce immondizia, cioè cose sbagliate, se ne ottiene immondizia anche in uscita". È il programmatore che fa fare al calcolatore il lavoro al quale è stato destinato. Se il programmatore è qualificato, il calcolatore sembra intelligente; se il programmatore non è così qualificato, il calcolatore non ha alcuna scelta se non fare *Gigo*.

Il software

Software è il termine globale per indicare i programmi gestiti da un calcolatore. Il software è il *conducente* del calcolatore; senza di lui il calcolatore non funziona. Il software è paragonabile alle informazioni di cui il cervello umano ha bisogno per prendere decisioni significative. L'hardware con la sua potenza – il cervello – è lì, ma senza esperienza. Esperienza è la parola che definisce *la condizione di chi ha provato qualche cosa di simile precedentemente, ha agito su di essa in modo tale da acquisire risultati positivi o no e ne ricorda gli stimoli di entrata e i risultati*. Senza esperienza il cervello non può fare un lavoro molto utile. Lo stesso dicasi per un calcolatore; attraverso il software dobbiamo dargli ordini perché sappia come *agire* quando incontra certe situazioni.

Sfortunatamente il calcolatore è molto meno intelligente del cervello; può decidere soltanto per il *sì* o per il *no* e anche per fare ciò sono necessarie *situazioni* molto specifiche. L'avverbio *forse* non è noto a un calcolatore, perché *forse* esprime un dubbio e un calcolatore non può esprimere alcuna sensazione di dubbio, poiché non ama, non odia e non ha pareri. Se dite a un calcolatore di fare qualcosa, tenterà di farla. Il calcolatore si attiene soltanto alla conoscenza del software, mai alla sua, dal momento che gli manca.

Un calcolatore può essere paragonato ad un impiegato che ha tutte le capacità potenziali di attenersi a un complesso di direttive, ma non le ha ancora ricevute. Le direttive consentono all'impiegato di risolvere un problema e pertanto sono assimilabili al software che pone il calcolatore in condizione di risolvere dei problemi. Però l'impiegato può *imparare*, vale a dire che dopo poche volte, non abbiamo bisogno di dargli le stesse direttive, perché le ricorda. Il calcolatore non

può *imparare* e perciò dobbiamo dargli ordini ogni volta che vogliamo faccia qualcosa, indipendentemente dal fatto che abbia già assolto gli stessi compiti migliaia di volte.

Una delle principali caratteristiche del cervello umano sta nella sua capacità di riconoscere i modelli. Se io ti conosco, io ti riconoscerò indipendentemente dal fatto che tu sia vestito in un modo o nell'altro, o che guardi in una direzione o in un'altra. I calcolatori non sono in grado di fare cose del genere. Molti progetti attualmente in corso, finalizzati all'intelligenza artificiale, sono focalizzati sulla realizzazione di macchine in grado di riconoscere i modelli. In questo campo non si sono fatti molti progressi, ma gli studi continuano. Le cose che i calcolatori fanno – in effetti molto meglio del cervello umano – si trovano nelle aree della logica deduttiva e del richiamo dei dati ed è per questo che sono così utili all'uomo.

Possiamo dividere il software in quattro categorie: software di sistema, programmi biblioteca, programmi già pronti, programmi sviluppati per l'utente. Nelle sezioni che seguono tratteremo brevemente ciascuna categoria.

Software di sistema

Questa parte di software è necessaria per far funzionare il sistema generale. Appartengono a questa categoria il sistema operativo e i *linguaggi* con i quali il programmatore può esprimere i propri ordini al calcolatore.

Come ho ricordato precedentemente, il sistema operativo o di controllo si accerta che il sistema funzioni; rappresenta il legame fra l'hardware e gli utenti del sistema. Sotto il suo controllo ogni linguaggio di programmazione dispone del proprio *software di sistema*, normalmente diviso in due parti: i compilatori del linguaggio e i sistemi *run-time*. Quando il programmatore scrive un programma (denominato anche *codice* o *istruzioni del programma*), il compilatore del linguaggio traduce il codice nel linguaggio proprio della macchina, controllando al tempo stesso errori di sintassi. Si deve tener presente che il compilatore non controlla gli errori di logica del programma, bensì solo quelli di sintassi. Il programma, compilato senza errori, può essere *caricato* nella memoria ed *eseguito* o *condotto* sotto il controllo del sistema *run-time* del linguaggio di programmazione. Di nuovo, non si

possono eseguire controlli di errori di logica. Se il programma dà i risultati desiderati dipende rigorosamente dal programmatore.

Alcuni linguaggi combinano compilatore e sistemi *runtime* in un segmento, ma il costo dell'esecuzione di programmi in quei tipi di linguaggio è normalmente più elevato che nei linguaggi che separano il compilatore dal segmento *runtime*.

Biblioteca di programmi

Questi programmi sono *proprietà della comunità* di cui fanno parte gli utenti del calcolatore. Si tratta di soluzioni generalizzate per vari problemi. Spesso i programmatori hanno bisogno di algoritmi generali nei loro programmi, ma anziché riscrivere sequenze di istruzioni codificate ogni volta che si rendono necessarie, il programmatore può includere codici biblioteca prescritti all'interno del suo programma, con risparmio di tempo e di denaro.

Nelle biblioteche di sistema si possono trovare anche dei giochi. È divertente vincere il calcolatore in una partita ed ora i calcolatori hanno addirittura dei tornei di scacchi propri, nei quali *giocano* uno contro l'altro.

Esempi di programmi biblioteca tipici sono dati da sequenze di istruzioni codificate matematicamente quali i programmi per la determinazione di curve di determinata caratteristica (vds. curva esponenziale) che meglio si adattano a un certo numero di punti del piano e le equazioni polinomie, sequenze di istruzioni codificate relative, ad esempio, al calcolo delle probabilità e all'analisi infinitesimale, formule di ingegneria, programmi commerciali.

Spesso nelle biblioteche di programmi computerizzati si possono trovare anche sequenze di classificazione generalizzata. La disposizione dei dati in sequenza ascendente o discendente è un compito richiesto frequentemente ai calcolatori. Di solito anche se la suddetta disposizione non è considerata il requisito fondamentale, ci si aspetta che la richiesta di situazioni di retribuzioni del personale, per settore, sia corredata da una situazione ordinata anche in base al nome e all'ammontare della retribuzione. Quasi tutti i programmatori che conosco, prima o poi vengono da me per mostrarmi magnifiche sequenze ordinate di istruzioni codificate, appe-

na scritte. In effetti, gran parte di queste sequenze *reinventate* sono valide, il problema è che sono reinventate, e ciò equivale a una perdita di tempo. Le biblioteche di programmi sono là per essere usate. I programmatori dovrebbero essere a perfetta conoscenza di tutti i programmi disponibili nella biblioteca del loro calcolatore. Quei programmi stanno al calcolatore, come la teoria del volo sta ai costruttori di aerei. Tutti i costruttori di aerei, infatti quando costruiscono un nuovo apparecchio non devono reinventare le formule dell'aerodinamica, ma si basano su tecnologie note e già sperimentate.

Programmi preconfezionati

Si tratta di sistemi utilizzabili dai non-programmatori, quali ad esempio i pacchetti di analisi statistiche, i sistemi di programmazione progetti, di modellistica finanziara, di gestione di base dati.

I *pacchetti* statistici, quelli per la programmazione di progetti e i finanziari consentono all'utente di risolvere problemi ricorrendo a codici di programmi prescritti. L'utente è condizionato dal tipo e dal formato di entrata e uscita consentiti da questi sistemi, ma spesso questo non è un problema. Quando possibile, l'uso di questi pacchetti risulta vantaggioso; se per soddisfare le esigenze dell'utente si rende necessaria qualche modifica, la durata del tempo di programmazione è spesso ridotta fino al 90 per cento; il che assicura riscontro più veloce, costo inferiore e maggiore attendibilità.

Nello sviluppo di questi sistemi preconfezionati si è proceduto a grandi passi, così che per dare all'utente il servizio richiesto, spesso il programmatore non è necessario.

Lo sviluppo dei sistemi di gestione delle basi dati è in continua evoluzione. Spesso i programmi commerciali si occupano dell'ingresso dei dati sotto certe forme predeterminate, del calcolo dei risultati e dell'uscita delle relazioni con variazioni di ogni genere. Nel passato, ogni volta che si doveva scrivere un nuovo programma di questo tipo, i programmatori dovevano *reinventare la ruota;* si doveva programmare un nuovo design della base dati, scrivere la logica per tutti i calcoli e sviluppare le specifiche dei rapporti. Lo sviluppo del design del rapporto d'uscita richiedeva di solito molto tem-

po, ed era molto difficile se non impossibile, modificare i formati di uscita una volta definiti e programmati. Con lo sviluppo dei sistemi di gestione delle basi dati generalizzate, questa situazione si è modificata radicalmente.

Il sistema di gestione della base dati è un programma con logica predeterminata e generalizzata incorporata; tutto ciò che il programmatore deve fare è di dire al sistema quali tipi di campi ciascuna delle sue registrazioni di dati dovrebbe comprendere e quale sarà la loro interrelazione. Poi il sistema predisporrà il *file* per il programmatore e farà tutta la *contabilità* necessaria per consentire aggiornamenti della base dati e per facilitare l'acquisizione di rapporti significativi. Lo sviluppo del sistema della gestione della base dati ha modificato i compiti della programmazione commerciale del giorno d'oggi.

L'esempio che segue illustra la facilità con la quale si può scrivere un rapporto di gestione di una base dati. Supponiamo di voler scrivere un rapporto che ordina in sequenza ascendente le retribuzioni dei dipendenti di un'impresa, le cui registrazioni sono immagazzinate in forma permanente in un *file* di dati. Successivamente, vogliamo indicare un aumento del 10 per cento sulle retribuzioni. In linguaggio di programmazione convenzionale, tale programma richiederebbe probabilmente circa 15 o più istruzioni al calcolatore, ma nel sistema di gestione della base dati, tutta l'elaborazione normalmente può essere eseguita con appena due frasi:

Scegliere tutti (vengono scelte tutte le registrazioni relative ai dipendenti che si trovano nel *file*)
Stampare/e ordinare le retribuzioni = sequenze ascendente/nome, retribuzione + 0,1* della retribuzione.

Con la seconda frase si chiede al calcolatore di stampare il nome di ciascun dipendente scelto nel primo ordine, dopo che i nomi sono stati ordinati in sequenza ascendente. Si chiede inoltre al calcolatore di stampare la retribuzione + 0,1* della retribuzione stessa, vale a dire che il calcolatore stamperà per ogni individuo il risultato dalla somma della retribuzione più il 10 per cento.

Le due frasi precedenti sono ordini dati in uno dei sistemi più comuni di gestione di base dati. Ovvio che potete capire questi ordini, senza aver mai imparato un linguaggio di pro-

grammazione. I sistemi di gestione di una base dati consentono ai programmatori di *parlare* ai calcolatori in un linguaggio che è molto simile alla lingua parlata, anche se gli ordini sono molto schematici e rigidi per natura. Ma l'aspetto più importante, è che questi nuovi *linguaggi* consentono di prendere lo stesso tipo di *scorciatoie* cui siamo abituati nella vita quotidiana. I termini *stampare e ordinare* sono parole significative e comprensibili per tutti noi, mentre se vi facessi vedere il programma in un altro linguaggio, con le stesse operazioni, sembrerebbe tutto molto più complicato.

Con l'evoluzione della gestione delle basi dati il manager, che non ha avuto occasione di imparare a comunicare con i calcolatori mediante linguaggi di programmazione convenzionale, ha a disposizione un valido strumento. Spesso ha ora la possibilità di scrivere i propri programmi, senza bisogno di dover spiegare prima il suo problema e la soluzione desiderata a un designer di sistemi, che a sua volta dovrebbe spiegare a un programmatore come scrivere il programma. Con l'avvento della programmazione della gestione delle basi dati è molto più facile e spesso più produttivo che un manager impari a programmare, che per un programmatore imparare il complesso delle attività di un manager. Ciò non significa che i programmatori si estingueranno; l'esecuzione di grandi sistemi e il rafforzamento di sistemi più piccoli diversificati rimarranno il campo dei programmatori. Peraltro, l'evoluzione del sistema di gestione delle basi dati continua, vi è la tendenza di *italianizzare* quanto più possibile le strutture dei comandi. Oggi ordini quali se *il nome è Rossi allora stampare la retribuzione*, oppure *selezionare insoluti superiori a 150.000 lire*, fanno già parte dei sistemi di gestione delle basi dati. Il microcalcolatore in casa vostra, nel prossimo futuro, impiegherà linguaggi di gestione delle basi dati; cosicché sarete nella condizione di avere una risposta che vi salverà magari la vita, quando dattiloscriverete se *il veleno è ammoniaca stampare cura*.

I sistemi di programmazione-progetti, denominati anche sistemi di gestione progetti, vengono usati per programmare e controllare grandi quantità di compiti necessari per completare un *progetto*. Qualsiasi progetto con più di 50 compiti connessi è un buon candidato all'approccio della programmazione di progetto. Il programma elenca i compiti a seconda delle relazioni che intercorrono fra i compiti stessi, sulla

base di quanto definito dall'utente. Ad esempio, dovendo costruire una casa, il compito della verniciatura dei davanzali delle finestre è in relazione con il montaggio delle finestre. Di solito, si pittura dopo la posa in opera. Inoltre, la pulitura degli schizzi di vernice capitati involontariamente sulle finestre sarà fatta dopo la verniciatura. Però, potremmo non pulire le finestre subito dopo la verniciatura: la vernice deve asciugarsi prima di poterla raschiare via. In un sistema di programma di progetto, enunceremmo questo problema mediante un ordine del tipo: *la pulitura delle finestre può iniziare un giorno dopo il completamento della verniciatura.*

L'esecuzione di programmi, come la costruzione di una casa, la stesura di un sistema di un calcolatore, o le correlazioni relative all'installazione di stazioni di superficie per la ricezione via satellite, richiede molti compiti, caratterizzati anche da molte interazioni. Spesso, non si tratta solo di relazioni da compito a compito successivo, a compito successivo, bensì di *reti* complicate, nelle quali le fasi hanno in comune molte relazioni. I sistemi di programmazione dei progetti mettono l'utente nella condizione di esaminare i progetti pianificati per determinare la totale fattibilità nei limiti di tempo desiderati. Gli aggiornamenti delle proiezioni di un progetto, nel corso dell'esecuzione dei compiti, consentono ai manager di essere sempre ed esattamente al corrente della parte del progetto completata e di confrontare l'andamento reale dei lavori con il programma originale.

Molti sistemi per la gestione progetti, oltre a offrire questo tipo di programma su una base *relazione e durata dei compiti*, possono incorporare negli algoritmi della programmazione la componente delle limitazioni delle risorse. I programmi di questo tipo trattano della disponibilità delle risorse in termini di personale, denaro e macchine per l'adempimento dei compiti finalizzati a determinati progetti. Quasi mai si dà il caso di una disponibilità illimitata di risorse. È noto che la data entro la quale un determinato progetto deve essere completato dipende molto dalle risorse che si rendono necessarie. L'inserimento del programma della limitazione delle risorse in quello generale dei compiti che configurano un progetto, mette in evidenza problemi connessi con l'esecuzione dei compiti, per cui le soluzioni proposte

possono essere immediatamente valutate a fronte dei risultati che probabilmente produrranno nel rimanente periodo della vita del progetto.

Nel quadro dei vari sistemi per la gestione progetti, oggi disponiamo anche delle informazioni sui costi dei progetti in corso. Oltre a fornire l'ammontare di spesa aggiornato di un progetto, le informazioni sui costi permettono al manager di valutare immediatamente le modifiche del quadro finanziario di un progetto, nell'ipotesi di una fluttuazione dei costi, durante lo stato di avanzamento del progetto stesso. Anche i costi connessi con i vari compiti di un progetto possono essere considerati come *risorse*. La programmazione di un progetto informata al criterio della disponibilità del denaro dà al manager un quadro molto chiaro sul *cash flow* necessario.

Ora le modifiche a un progetto nel bel mezzo del suo sviluppo, non devono impedirne la comprensione. Il progetto sarà sotto controllo mediante regolari aggiornamenti; la restante durata, i costi e le esigenze in termini di risorse saranno fattori tutti ben noti al *project-manager*. I sistemi di gestione dei progetti consentono processi decisionali più veloci e più accurati. I compiti fastidiosi della riprogrammazione di un progetto in continua evoluzione e della riconvalida delle mutate relazioni sono lasciati al calcolatore.

Programmi sviluppati per l'utente

L'ultima categoria di software dei sistemi è quella dei programmi sviluppati per l'utente, che costituiscono ancora la maggior parte del carico del lavoro giornaliero dei calcolatori. Sono costruiti dal basso verso l'alto. Si tratta di programmi analizzati da analisti, progettati da designers, programmati da programmatori che per scriverli usano linguaggi complessi.

Questi programmi risolvono problemi in aree molto differenziate: libri paga, balistica dei razzi, previsioni meteo automatizzate, calcolo del controllo dell'emissione partitario generale, circolazione libraria ed anche programmi coi quali scrivere programmi. Queste sono alcune fra le migliaia di applicazioni di software per le quali i calcolatori vengono programmati ogni giorno.

Il programmatore

Prima di trattare del programmatore, vediamo cos'è un programma.

Il programma è un insieme di istruzioni date a un calcolatore. È come una lista della spesa: mezzo chilo di spinaci; pane di tipo speciale (se il negozio non ne ha più, comperare panini per hamburger); un sacchetto di patate: vale a dire un elenco di istruzioni su ciò che si deve fare. L'unica differenza fra questa lista e un programma è che l'ordine della lista della spesa può essere *eseguito* o compiuto secondo qualsiasi successione che meglio si presenta all'acquirente quando è al supermercato, mentre il programma di un calcolatore deve essere eseguito, una istruzione dopo l'altra, nella sequenza con cui le istruzioni si presentano sul programma. I programmi consentono di saltare da un'istruzione all'altra fuori sequenza, ma ciò deve essere una richiesta esplicita del programma.

Le istruzioni costituenti un programma sono scritte in *linguaggio* speciale che è un mezzo per comunicare con un calcolatore. Come nel mondo esistono molte lingue parlate e scritte, così vi sono molti linguaggi per computer. Alcuni sono stati ideati da costruttori di calcolatori per scrivere programmi specifici per le loro macchine; altri sono di natura più generica e si possono adattare, mediante piccole o grandi riprogrammazioni, a calcolatori molto diversi.

Oggi i linguaggi più comuni sono il BASIC (Beginners All – Purpose Symbolic Instruction Code), COBOL (Common Business Oriented Language), FORTRAN (FORmula TRANslation), APL (A Programming Language), ALGOL (ALGOrithmic Language) e PL/1 (Programming Language numero 1). Ciascuna ditta che vende calcolatori e include nei suoi linguaggi caratteristiche che gli altri costruttori forse non hanno, definisce il suo linguaggio un *superset* rispetto alla norma. Ne conseguono nomi come Super FORTRAN e COBOL *Extended*. Sono stati istituiti dei comitati per cercare di fissare degli standard per i vari linguaggi, in modo che il linguaggio A del costruttore X sia identico al linguaggio A del costruttore Y. In quest'area si sono compiuti dei progressi. In sede di pratiche per gli approvvigionamenti, il governo statunitense obbliga le ditte venditrici di calcolatori e di servizi computerizzati a standardizzare le loro offerte di linguaggi.

Probabilmente è l'unica istituzione che può imporre un tale obbligo; le gare per approvvigionamenti alle quali costruttori e organizzazioni di servizi fanno offerte sono così rilevanti da ritenere ragionevole che i venditori investano tempo, sforzi e denaro per cambiare i loro software, in vista del soddisfacimento di tali esigenze. Il National Bureau of Standars (NBS) e l'American National Standards Institute (ANSI) sono le organizzazioni governative trainanti nell'area della standardizzazione dei software dei calcolatori.

Molto in fretta vi insegnerò il BASIC, per dimostrarvi che i rudimenti della programmazione sono molto facili da imparare. Supponiamo di voler scrivere un programma che calcoli il risultato del seguente problema aritmetico:

$$1 + 1 =$$

Perché il calcolatore ci risolva questo problema dobbiamo scrivere l'espressione in modo che lui la comprenda. Se ricordate quanto è stato detto della memoria sapete che possiamo *indirizzare* parole alla memoria del calcolatore. Occorre tener presente poche cose.

1. La memoria è come una fila di cassette per le lettere di un condominio. Si può paragonare una parola a una cassetta per lettere. La cassetta ha un *indirizzo*, in modo che il postino sa quali lettere vanno in quella data cassetta. Quando il portalettere deposita le lettere nella cassetta, questa ha un *contenuto* ovvero, in lingua computerizzata, l'indirizzo ora ha un *valore*.

I calcolatori operano come i portalettere: ricercano l'indirizzo esatto, poi mettono qualcosa nella cassetta. La differenza fondamentale sta nel fatto che le cassette per le lettere del condominio hanno già indirizzi predeterminati, mentre il programmatore di un calcolatore quando scrive il suo programma può definire denominazioni convenzionali sue proprie. Ancora: a differenza del postino, il programmatore non ha bisogno di sapere dove gli indirizzi delle cassette sono fisicamente ubicati; per spedire la posta non ha bisogno di sapere la città di destinazione, perché il calcolatore instraderà automaticamente il contenuto deliberato alla esatta posizione, una volta che questa sia stata fissata e depositata lì.

2. A conoscenza di ciò, vi è soltanto un'altra cosa che dobbiamo ricordare prima di scrivere il nostro programma:

il segno di uguale (=) assume significato diverso da quello che comunemente gli diamo. Nei linguaggi computerizzati questo è un segno di *assegnazione* che significa: quando mettete qualcosa nella vostra cassetta per lettere specificando Cassetta = 1, significa che ora i contenuti della cassetta sono 1, sostituendo qualsiasi evento che era nella cassetta prima di questa transazione. Perciò: Cassetta = 1 non significa uguagliare la Cassetta della posizione con 1, che vorrebbe dire cambiare l'indirizzo, significa che il *valore* della vostra Cassetta ora è diventato 1.

Vi prego di rileggere i tre paragrafi precedenti; quando ne avrete completamente acquisito i concetti di base, sarete sulla buona strada per diventare programmatore.

Ma torniamo al nostro programma BASIC. Perché il calcolatore ci risolva il nostro problema, abbiamo bisogno di almeno un indirizzo nella memoria nel quale immagazzinare un valore. Chiamerò quell'indirizzo V (per mia propria scelta: potrei chiamarlo in qualche altro modo). *Prenoto* questo indirizzo designandolo al calcolatore in un ordine di designazione. L'ordine ha questa configurazione:

PONIAMO V = 1

Ora sappiamo che ciò significa: "Dare una cassetta per lettera, chiamarla V e immagazzinare nella cassetta il valore 1".

Adesso possiamo risolvere il nostro problema nel seguente modo:

PONIAMO V = V + V

che significa: "Mettere nell'indirizzo V il valore che al momento risiede in V, aggiunto al valore che al momento risiede in V". Come è noto, il valore che risiede in V era 1; l'ordine di cui sopra dice al calcolatore di aggiungere 1 a 1 e di sostituire il valore che esisteva in V (cioè 1) con il risultato, cambiando, dopo l'ordine, i contenuti dell'indirizzo V in 2.

Detto questo vi è poco da aggiungere. Manca una cosa: ora il calcolatore conosce la risposta (ha messo il valore 2 in V), ma voi no, perché non avete chiesto al calcolatore di stampare per voi il risultato. Ricordate che un calcolatore è uno dei soggetti più taciturni che esistano: se non gli dite di fare qualcosa, non lo farà da solo. Il risultato deve essere stampato. Il metodo per risolvere problemi del genere differisce

considerevolmente da un linguaggio all'altro, ma nel BASIC funziona così:

STAMPARE "IL VALORE ATTUALE DI V = ", V

Questo ordine per la stampa dice: stampare per me il valore attuale della parola V, dopo aver stampato la frase:

IL VALORE ATTUALE DI V =

Come vedete, le frasi possono essere stampate prima o dopo che il calcolatore stampi il valore della parola stessa. Il calcolatore sa che ci si aspetta che lui stampi una frase del tipo.

IL VALORE ATTUALE DI V =

nel momento in cui il programmatore mette questa espressione *letterale* fra virgolette.

Adesso abbiamo scritto un programma BASIC. Ripeto il tutto:

```
PONIAMO V = 1
PONIAMO V = V + V
STAMPARE "IL VALORE DI V = ", V
FINE
```

L'ordine FINE dice al calcolatore che il vostro programma è finito.

Spero con ciò di riuscire a eliminare la mistica dai calcolatori. Se decidete di impadronirvi maggiormente della programmazione, non aspettatevi di incontrare qualcosa di più difficile. È semplice: quelle difficoltà non esistono.

A scuola, di solito ero molto preoccupato quando studiavo su libri di testo, con titoli come *Introduzione all'analisi statistica*. Regolarmente si trattava di libri di testo di centinaia di pagine.

Il pensiero che un voluminoso manoscritto fosse appena un'introduzione (chi sa quali epistole sarebbero seguite) mi scoraggiava enormente dal fare ricerche sull'argomento. Avreste dovuto avere una esperienza simile. Ecco la sorpresa: se avete seguito la trattazione fatta fin qui, adesso siete dei programmatori. Non sarete ancora buoni programmatori, perché non avete alcuna esperienza e non conoscete tutti gli ordini disponibili nei vari linguaggi computerizzati, ma

nondimeno siete dei programmatori.

Quanti ordini di programmazione ci sono nel linguaggio di un calcolatore? Non molti davvero. Dipende dalla sofisticazione del linguaggio, ma di norma i linguaggi non hanno più di 30 o 40 tipi di ordini differenti. Ora avete visto gli ordini STAMPARE e FINE nel linguaggio BASIC e siete in grado di imparare altrettanto facilmente gli altri ordini disponibili.

Qualche parola ancora sugli indirizzi della memoria. Spesso un programmatore farà riferimento al *nome variabile*. Ciò che un programmatore intende dire con questa espressione è *il nome che ho dato alla posizione della memoria (indirizzo) nella quale immagazzinerò un valore (essendo questo variabile, in relazione a ciò che voglio mettere lì di quando in quando*). Così, il termine *nome variabile* non significa niente di più del nome dell'indirizzo nel quale verranno immagazzinati i valori. L'attribuzione del nome da parte del programmatore a questo indirizzo è sotto il suo controllo, nonostante che molti linguaggi abbiano delle restrizioni quanto meno sulla lunghezza del nome. Alcuni linguaggi consentono solo un nome a un carattere (che confonde qualcuno che legge il programma); altri linguaggi, quali il FORTRAN, consentono fino a sei caratteri per i nomi delle variabili; altri ancora, quali il COBOL, consentono 30 e più caratteri per dare il nominativo a un indirizzo. Altra restrizione consiste nel fatto che certi nomi sono *riservati*, vale a dire che questi nomi sono elementi del linguaggio stesso e pertanto non possono essere usati come nomi di posizioni della memoria.

Troverete che i programmi scritti bene sono facili da leggere. I nomi variabili usati descriveranno il loro scopo il più chiaramente possibile. Considerate, per esempio, il seguente ordine COBOL:

SOTTRARRE SPESE DA VENDITE LORDE
ESPONENDO VENDITE NETTE

SOTTRARRE è un *verbo* COBOL o dell'operatore; SPESE è un nome variabile, così denominato dal programmatore; DA è un *operatore* COBOL; VENDITE LORDE è un nome variabile; ESPONENDO è di nuovo in COBOL; VENDITE NETTE è ancora un nome variabile inventato dal programmatore. Sono certo che ammetterete che questo ordine è di facile lettura

ed è chiaro cosa ho voluto fare. Lo stesso ordine avrebbe potuto esser scritto così:

PONIAMO A = B − C

dove A sta per VENDITE NETTE, B sta per VENDITE LORDE e C sta per SPESE. I risultati del calcolatore sarebbero esattamente gli stessi, perché il calcolatore non potrebbe curare meno ciò che voi nominate vostre variabili; ma per noi uomini, questa affermazione è molto meno autoesplicativa di quella precedente.

Eccovi la definizione di un *buon* programma: (1) il programma *funziona*, il che significa che assolve i compiti per i quali era stato designato e (2) il programma è ben documentato, il che significa che un non-programmatore può seguire la logica del programma facilmente, mediante la semplice lettura degli ordini del programma e i commenti.

Se siete nella condizione di avere programmi scritti per voi, insistete sulla buona documentazione. In genere i programmatori non documentano molto bene i loro programmi, ma il tempo extra necessario per frammischiare spiegazioni con righe di codice (che ogni linguaggio consente) risparmierà a voi e al programmatore soldi e tempo. Niente di peggio che tentare di mantenere, cambiare o accrescere un programma scritto da qualcun altro, se il programma non è documentato. Al meglio, programmi non documentati sono una forma di sicurezza del lavoro: "Tu mi fai fuori e staremo a vedere che cosa succede ai tuoi programmi", al peggio, codici non documentati possono, almeno per un momento, obbligare alla totale rigidità di un'impresa, che si tradurrebbe facilmente nel deterioramento dei profitti con conseguenti effetti negativi sui posti di lavoro.

Non è mia intenzione essere burbero, ma perché ammettere programmi malamente documentati? È come costruire un aeroplano senza scrivere il manuale della manutenzione: fino a che vola, tutto è magnifico, ma non appena qualcosa va male, è finita.

Oltre ad essere ben documentato, ogni programma degno di questo nome, dovrebbe avere una struttura chiaramente definita. Ciò significa che le sequenze logiche degli eventi dovrebbero essere scritte in modo tale da essere facilmente comprese da un non-programmatore. La struttura di base è

la seguente:

Entrata (dati al calcolatore)

Calcolo (risultati desiderati)

Uscita (risultati)

dove ogni *funzione* è quasi un programma di per sè e porta a termine i risultati desiderati.

Ecco parte di un programma COBOL funzionalmente progettato e facile da seguire:

```
*
*** SEZIONE ENTRATA
*
    ACCETTARE VENDITE LORDE
    ACCETTARE SPESE
*
*** SEZIONE CALCOLO
*
    SOTTRARRE SPESE DA VENDITE LORDE ESPO-
    NENDO VENDITE NETTE
*
*** SEZIONE USCITA
*
    DISPLAY VENDITE NETTE
```

Di quando in quando, abbiamo notizia di un calcolatore che ha commesso un errore conclusosi con la spedizione di un assegno di 2 milioni di dollari a un pensionato. La cosa offre materia di grande interesse per i giornali, ma attribuisce ai calcolatori cattiva stampa inutilmente. Come regola, i calcolatori non commettono errori, ma la stessa affermazione non può estendersi ai designer e ai programmatori. Errori di questa natura possono farsi risalire molto spesso ad un errore nella logica del programma, o a interrelazioni scorrette di programmi nell'ambito di un sistema, per cui una situazione particolare viene trattata non correttamente.

Ritorniamo al nostro programma di prima. Supponiamo che invece di specificare SOTTRARRE SPESE DA VENDITE LORDE ESPONENDO VENDITE NETTE il programmatore abbia scritto SOTTRARRE VENDITE LORDE DA SPESE ESPONENDO VENDITE NETTE. Il risultato non sarà cor-

retto, ma il calcolatore non lo sa. Il calcolatore fa solo ciò che il programmatore gli dice di fare. Un buon programmatore proverà il suo programma completamente e si accerterà che tutti gli ordini e il *flusso* generale del programma siano corretti.

Nell'esempio sopra riportato qualsiasi programmatore probabilmente si accorgerebbe molto presto che qualcosa è andata storta. Sfortunatamente i programmi non sono sempre così semplici, per cui esistono molte possibilità di commettere errori. Consideriamo un'ipotesi riferita al libro paga Nel caso più semplice un lavoratore viene pagato su base settimanale, tasse e oneri per l'assistenza socio-sanitaria gli vengono calcolati e trattenuti, viene stampato l'assegno. Ma questa non è *vita reale*. In effetti quel dipendente la settimana scorsa è stato malato un giorno, sua moglie ha dato alla luce un figlio (la qual cosa modifica gli assegni familiari) e intende contribuire al fondo pensioni dell'impresa per un ammontare del 5 per cento del suo reddito. E ancora, per poche sere, ha fatto dello straordinario – calcolato in modo diverso – e ha lavorato di domenica, per recuperare un giorno di lavoro perduto la settimana prima. Per complicare la faccenda, quel dipendente aveva fatto domanda, che era stata accettata, per un rimborso relativo a un corso che aveva frequentato e questo rimborso doveva entrare nella busta paga di questa settimana, ma l'ammontare non doveva superare il tetto del rimborso annuo previsto. Come vedete, posso continuare ancora; le occasioni di errori nell'ambito di un programma sono infinite.

Come può un programmatore provare tutto? È necessario disporre di buone procedure di prova con dati della vita reale, buona immaginazione da parte del programmatore e tempo. Come regola, il tempo di prova è o dovrebbe essere più lungo di quello della programmazione. I programmatori chiamano questa operazione *spulciatura* vale a dire mandar via le pulci dal programma. La frase *nel programma c'è una pulce* ebbe origine all'epoca in cui i calcolatori erano raffreddati ad acqua ed occupavano due o tre piani. Topi e altri insetti qualche volta facevano una capatina dai macchinari procurando l'arresto repentino di tutta l'operazione. Così, il calcolatore doveva essere *spulciato*. Di norma i topi non vanno più nei calcolatori, ma gli *insetti* si ficcano ancora nei programmi. Questo argomento è trattato più diffusamente

nella sezione "Prova programmi", del Capitolo 3.

Ciò che rende la programmazione un po' infida è la sequenza logica degli ordini. Se volete che il vostro programma faccia determinati passi, questi devono essere scritti in progressione logica. È come cambiare una gomma: non mettete su la nuova se prima non avete tolto la vecchia. Ovvero, più nel dettaglio, i passi per cambiare una gomma sarebbero i seguenti. Per prima cosa aprite il bagagliaio. Poi portate la gomma nuova e gli attrezzi necessari fuori dal bagagliaio. Poi mettete il girabacchino nella dovuta posizione e sollevate un po' la macchina. Poi allentate i dadi della ruota da sostituire. Poi sollevate la macchina finché la gomma non è più a contatto col terreno e svitate completamente i dadi. Rimuovete la ruota e sostituitela con quella nuova. Stringete i dadi, fate scendere la macchina quasi del tutto, avvitate fino in fondo i dadi e fate scendere la macchina completamente. Disinserite il girabacchino, mettete il coprimozzo sulla ruota nuova, riponete la gomma vecchia nel bagagliaio e andate dal gommista perché vi ripari la gomma.

Se aveste eseguito alcune di quelle azioni in una sequenza diversa vi sareste trovati in difficoltà. La stessa cosa vale per programmare un calcolatore.

Individui dotati di molta fantasia sono capaci di immaginare le conseguenze dovute alle varie fasi dell'*esecuzione* di un programma, perciò terranno gli occhi aperti per possibili condizioni *impossibili*.

In genere si crede che i programmatori debbano essere dei matematici. Fortunatamente questo non è vero. Qualche conoscenza degli elementi fondamentali di algebra è utile, ma molto più importante è la maniera logica di pensare. È opinione che chi è in grado di prendere una laurea in matematica pensa logicamente e pertanto ha la potenzialità per diventare un bravo programmatore. Ciò può essere vero, ma alcuni fra i migliori programmatori che conosco hanno precedenti di studio in discipline quali la letteratura, la storia o l'economia.

Come si fa per diventare *bravi* programmatori? Occorre fare molte programmazioni, che affinano la professionalità logica necessaria; e almeno una volta si devono commettere vari tipi di errori. Per imparare qualcosa non c'è modo migliore che sperimentare se un metodo funziona, oppure no. Non è sufficiente *aver sentito parlare* di un *metodo sbagliato*;

per sapere nella realtà ciò che è giusto e ciò che è sbagliato, bisogna *provare in proprio.*

Ogni giorno sui giornali si leggono annunci economici con richieste di programmatori cón tre, quattro, cinque o più anni di esperienza nel linguaggio XYZ per calcolatori ABC. Si tratta di annunci non necessariamente restrittivi. Infatti, si dovrebbero cercare programmatori con tre, quattro, cinque o più anni di esperienza: punto e basta. L'esperienza acquisita nei linguaggi e sulle macchine è incidentale. Un bravo programmatore imparerà un nuovo linguaggio nel giro di un mese o due o meno, ma è necessaria una pratica seria (e molto tempo) per raggiungere un buon livello.

C'era poi quel tale che era molto ansioso di ottenere un lavoro nell'area della programmazione. Quando gli fu chiesto se conosceva COBOL, rispose· "Perbacco, credo di sì. Qual'è il suo nome di battesimo?".

Allegato a Zerouno n. 1 del febbraio 1982
direttore responsabile Italo Cammarata